Publication Number 29-376R12

32-BIT FAMILY LOADER DESCRIPTION MANUAL

INTERDATA®
Subsidiary of PERKIN-ELMER
Oceanport, New Jersey 07757, U.S.A.

January 1977

PAGE RÉVISION STATUS SHEET

PUBLICATION NUMBER 29-376

TITLE 32-Bit Family Loader Description Manual

REVISION R12 DATE 1/77

PAGE	REV.	DATE	PAGE	REV.	DATE	PAGE	REV.	DATE
i	R11	4/76	A9-3	R11	4/76			
1	R10	2/76	A9-4	R11	4/76			
2	R10	2/76	A9-5	R11	4/76			
3	R10	2/76	A9-6	R11	4/76			
4	R10	2/76	A9-7	R11	4/76			
5	R10	2/76	A9-8	R11	4/76			
6	R10	2/76	A9-10	R11	4/76			
7	R10	2/76	A9-11	R11	4/76			
8	R10	2/76	A9-12	R11	4/76			
9	R10	2/76	A9-13	R11	4/76			
10	R10	2/76						
11	R10	2/76						
12	R10	2/76						
13	R10	2/76						
14	R10	2/76						
15	R10	2/76						
16	R10	2/76						
17	R10	2/76						
18	R10	2/76						
19	R10	2/76						
20	R10	2/76						
21	R10	2/76						
A1-1	R10	2/76						
A2-1	R10	2/76						
A3-1	R10	2/76						
A4-1	R10	2/76						
A4-2	R10	2/76						
A5-1	R11	4/76						
A5-2	R11	4/76						
A6-1	R10	2/76						
A6-2	R10	2/76						
A6-3	R10	2/76						
A7-1	R10	2/76						
A8-1	R12	1/77						
A8-2	R12	1/77						
A8-3	R12	1/77						
A8-4	R12	1/77						
A8-5	R12	1/77						
A8-6	R12	1/77						
A8-7	R12	1/77						
A8-8	R12	1/77						
A8-9	R12	1/77						
A8-10	R12	1/77						
A8-11/ A8-12	R12	1/77						
A9-1	R11	4/76						
A9-2	R11	4/76						

32-BIT FAMILY LOADER DESCRIPTION MANUAL

TABLE OF CONTENTS

1. INTRODUCTION 1

2. GENERAL INFORMATION 1

2.1 Types of Loaders 1
2.2 Loader Terminology 1
2.3 Object Program Formats 2
2.4 Summary of User Program Loaders 3

3. THE 32-BIT RELOCATING LOADER (03-067) 5

3.1 Features 5
3.2 Operation 6
3.3 REL Loader Bootstrapping 7

4. THE OS/32 LIBRARY LOADER 9

4.1 Program Loading Conventions 9
4.2 OS/32 Library Loader Operation and Commands 10
4.3 Library Loader Messages 16
4.4 Library Creation 18
4.5 Interface Between The Library And OS/32-ST 18

5. BULK/STORAGE BOOTSTRAP 19

5.1 Requirements 19
5.2 OS Selector 19
5.3 Loading Procedures 20

APPENDICES

APPENDIX 1 SUMMARY OF OS LIBRARY LOADER COMMAND A1-1/A1-2
APPENDIX 2 SUMMARY OF 32-BIT RELOCATING LOADER OPERATION A2-1/A2-2
APPENDIX 3 50 SEQUENCE LOADER A3-1/A3-2
APPENDIX 4 LOADER MEMORY MAPS A4-1
APPENDIX 5 USE OF THE BOOT PUNCHER A5-1
APPENDIX 6 INTERDATA 32-BIT LOADER FORMAT A6-1
APPENDIX 7 REVISION INFORMATION A7-1/A7-2
APPENDIX 8 RELOCATING LOADER LISTING A8-1
APPENDIX 9 DIRECT ACCESS BOOTSTRAP LOADER LISTING A9-1

ILLUSTRATIONS

Figure 1. 32-Bit REL Loader Operating Procedures 8
Figure 2. Memory Map (As Listed by the OS/32 Library Loader) 12

TABLES

TABLE 1. SUMMARY OF LOADER FEATURES 4
TABLE 2. MEMORY INITIALIZATION CONTENTS 20
TABLE 3. STANDARD BINARY INPUT DEVICE ADDRESSES AND OUTPUT COMMANDS 20
TABLE 4. STANDARD DIRECT ACCESS DEVICE ADDRESSES AND OUTPUT COMMANDS . . 21/22
TABLE 5. ERROR CODES 21/22
TABLE A6-1 CONTROL ITEM DEFINITION A6-2
TABLE A6-2 TAPE CODES A6-3/A6-4

32-BIT FAMILY LOADER DESCRIPTION MANUAL

1. INTRODUCTION

The primary purpose of this manual is to describe the features and operations of the OS/32 Library Loader (03-065) and the 32-bit Relocating Loader (03-067). In addition, various other loaders available on the 32-bit Family of Processors are described briefly for user orientation. Section 2 contains general information about loaders, and should be reviewed before reading other sections of the manual. Sections 3 and 4 describe the Relocating and OS/32 Library Loaders respectively.

2. GENERAL INFORMATION

2.1 Types of Loaders

The purpose of this section is to describe, to the INTERDATA user, the various loaders that are available on the 32-Bit Family of Processors, and to aid the user in selecting and using each loader program. There are four main classes of loaders.

1. 50 Sequence Loader - A memory resident loader that loads memory image data.

2. 32-Bit Relocating Loader - This is a stand-alone loader that loads object programs output by the INTERDATA 7/32 Assembler or Common Assembler (CAL), and does not require OS support.

3. OS Loaders - This class loads assembly language and FORTRAN programs under OS control.

   TYPES:

   OS/32-MT Resident Loader
   OS/32-ST Resident Loader
   OS/32-Library Loader
   OS/32-MT Task Establisher

4. OS/32-Bulk Storage Bootstrap Loader - This program acts as a front-end loader for OS/32-ST and MT resident on a bulk storage device.

2.2 Loader Terminology

The following list is a glossary of terms used in this manual which describe the loader features and program charactersitics.

ABSOLUTE PROGRAM	A program assembled to occupy fixed locations in memory.
RELOCATABLE PROGRAM	A program that can be loaded at a desired memory location specified at load time.
BIAS	The address specifying where a relocatable program is to be loaded.
LABEL	An eight character name that identifies a program.

32-BIT LOADER FORMAT	An encoding scheme which is used to represent object programs.
ZONED TAPE	A standard loader format tape that allows binary characters to be punched on the Teletype.
EXTRN	A symbolic reference to an external program or value linkable at load time.
ENTRY	A symbolic definition which allows reference by other programs to be resolved at load time.
COMMON	An area set aside by a loader at load time for reference by a number of programs.
UBOT	User Bottom of Core-UBOT is the first (lowest address) memory location available for loading user programs in an OS environment.
CTOP	Top of Core - The highest address of physical memory or user Task area.
BULK STORAGE	Disc, drum, magnetic tape or cassette.
IMPURE SEGMENT	A portion of a program declared at assembly time to be non-sharable between tasks.
PURE SEGMENT	A portion of a program declared at assembly time to be sharable (reentrant) between tasks (OS/32-MT)
IMPURE BIAS	Start location of the impure segment.
PURE BIAS	Start location of the pure segment.

2.3 Object Program Formats

INTERDATA 32-bit programs are normally supplied as binary object in various formats. These formats and the corresponding part number suffix are:

M11 32-Bit Loader Format Paper Tape
M21 32-Bit Loader Format Cassette
M31 32-Bit Loader Format Magnetic Tape
M51 32-Bit Loader Format Disc
M14 8-Bit Memory Image Paper Tape
M24 8-Bit Memory Image Cassette
M34 8-Bit Memory Image Magnetic Tape

Types M11, M21, M31 and M51 are INTERDATA 32-Bit Standard loader format as described in Appendix 6. INTERDATA 32-BIT LOADER FORMAT.

The 8-Bit Image tapes are loaded by using the 50 Sequence loader adjusted to read the correct amount of data. Normally, the 50 Sequence is modified for this purpose by changing the upper limit to the appropriate value. This tape format is used only for certain test tapes, such as the Processor Test and the 32-Bit REL Loader which cannot assume the presence of another loader program in memory.

2.4 Summary of User Program Loaders

All 32-Bit loader format tapes must be loaded by one of the following loaders.

32-Bit REL Loader	03-067
OS/32 Library Loader	03-065
OS/32-ST Resident Loader	03-075
OS/32-MT Task Establisher Task	03-077

Refer to Appendix 6 for a complete definition of the loader format. All of the above loaders load either zoned or non-zoned tapes.

Zoned tapes use only 4 bits of each tape frame for data with the other 4 bits assigned a "zone" code to make the entire frame a non-printing ASCII code. This format is used to punch tapes on an ASR Teletype tape punch; all INTERDATA programs produce this format when punched on a Teletype device. The non-zoned tapes use all 8-bits of each tape frame for data; therefore, these tapes are half as long as zoned tapes for an identical program.

Non-zoned paper tapes use an X'F0' as the first character of each record, enabling the loaders to identify the two formats. OS programs such as the Common Assembler (CAL) produce non-zoned tapes when punching to a High Speed Paper Tape Punch. This tape format results in less punching time, shorter tapes, and faster loading than with zoned tapes.

Each of these loaders provides a varying degree of loading capability. A brief description, in ascending order, follows:

32-BIT RELOCATING LOADER	Loads absolute or impure relocatable programs. No external symbolic referencing is allowed. This loader is mainly used to load Operating Systems or other stand-alone software, such as test programs.
OS/32-ST RESIDENT LOADER	Same features as the Relocating loaders except programs are loaded for execution under the OS. Pure relocatable segments are handled by this loader. For details on features and operation, see the OS/32-ST Programmer's Reference Manual, 29-380.
OS/32 LIBRARY LOADER	Includes features of the OS/32-ST resident loader and also resolves symbolic references/definition linkage (EXTRN/ENTRY) and processes program labels. Allows selective Editing of program libraries, building of load modules and overlays, and handling of common.
OS/32-MT TASK ESTABLISHER (TET/32)	Includes most of the features of the OS Library Loader but is specifically designed to establish tasks for OS/32-MT. The established tasks produced by TET/32 may be loaded only by the OS/32-MT Resident loader. For details on features and operation, see the OS/32-MT Programmer's Reference Manual, Publication Number 29-390.

Table 1 is a summary of loaders and their features.

TABLE 1. SUMMARY OF LOADER FEATURES

	32-Bit REL Loader	OS/32 Library Loader	OS/32-ST Resident Loader	OS/32-MT Resident Loader	OS/32-MT Task Establisher Task
ABS Programs	Yes	Yes	Yes	Yes	Yes
Impure REL Programs	Yes	Yes	Yes	Yes	Yes
Pure REL Programs	No	Yes	Yes	Yes	Yes
Program Labels	Ignored	Legal	Ignored	Ignored	Legal
ENTRYs	Ignored	Legal	Ignored	Ignored	Legal
EXTRNs	Illegal	Legal	Illegal	Illegal	Legal
Common Usage	Illegal	Legal	Illegal	Illegal	Legal
Error Handling	Halts	Typed Out	Typed Out	Typed Out	Typed Out
OUT Mode	No	Yes	No	No	Yes
MAP Facility	No	Yes	No	No	Yes
Library Handling	No	Yes	No	No	Yes
Library Building	No	Yes	No	No	Yes
Overlay Building	No	Yes	No	No	Yes
Automatic Post-Load Transfer	Yes	No	No	Yes	No
OS/32-MT Task Building	No	No	No	No	Yes
System Requirements	≈ 1.2KB	≈ 5KB+OS/32 -ST or OS/32-MT	OS/32-ST	OS 32-MT	≈ 16KB+OS/32-ST or OS/32-MT

3. THE 32-BIT RELOCATING LOADER (03-067)

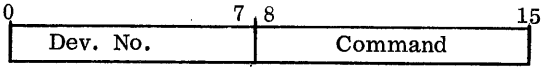
3.1 Features

The REL Loader contains drivers for Teletype, High Speed Paper Tape Reader, Intertape (Cassette) and Magnetic Tape devices. The following sections describe its features.

1. The input device for loading is defined by the Binary Input Device in the Device Definition Table in low memory. A device number of X'n2' is interpreted as a TTY, X'n3' as a High Speed Paper Tape device, and X'n5' as a Magnetic Tape or Cassette device.

   This halfword for various devices is shown below:

   Location X'78'

   Examples:

Teletype	0294
High Speed Paper Tape Reader	0399
High Speed Reader/Punch	1399
Magnetic Tape	85A1
Intertape (Cassette)	45A1

2. When reading binary data from Paper Tape, leader and illegal characters are skipped. Formats are discussed in Appendix 6.

3. Checksums and sequence numbers are checked after each binary record is read. Appropriate error halts occur when errors are detected.

4. The REL Loader is provided in relocatable bootstrap form. When loaded by the 50 Sequence the REL Loader is placed into the top of available memory (CTOP-X'500'). If more than 256KB is on the system, the loader will reside at X'3FB00'. Once the Loader is in the top of memory, any program can then be loaded to any other point in memory.

5. The first location (ORG) of the REL Loader is the starting location.

6. In the REL loader, Display Panel Indicators 0-15 are used to identify the meaning of loader halts. The light patterns used are:

0000	normal end
0001	checksum, parity error, or device unavailable
0002	sequence error
0003	attempt to load over loader
0004	reference-definition loop
FFNN	Illegal loader item encountered (NN = improper loader item)

   Parity errors detected while reading from Magnetic Tape or Cassette cause the REL loader to backspace and retry the operation five times. After the fifth attempt, the record is backspaced and a checksum/parity error pattern is displayed on the Display Panel. Depression of the RUN switch causes the record to be re-read. There is no provision for bypassing a record with a parity error. If the device is unavailable (off-line), the checksum/parity error pattern is also displayed.

7. The REL loader has the capability of selectively loading programs from magnetic tape and cassette libraries. Programs (separated by file marks) are specified by entering into the low order 8 bits of the data register of the Display Panel the number of file marks to be skipped by the Loader. When started, the REL Loader reads the Display Panel, skips the specified number of file marks, and begins loading immediately. Care should be taken to avoid inadvertant skipping. When loading from Paper Tape devices, the Display Panel is ignored. File skipping is not provided with Magnetic Tape and Cassette when the system is configured with Turn Key Panel.

8. The Loaders transfer immediately to the program loaded, if a transfer address is on the object medium.

When the REL Loader is executed at its starting address (ORG), the bias value is set to X'A00'. This bias value is used during program loading to adjust any relocatable data values. Note that absolute programs are stored at the absolute location specified for the data. Relocatable programs are stored from the location indicated by bias upward into memory. After a program has been loaded, the bias value is adjusted to point to the next available location. To indicate that the load is complete, the Loader halts (if no transfer address is present) with the Wait light ON, and with 0000 contained in the Display Register. At this time, the adjusted bias value is held in Supervisor Register 0. (This register may be displayed on the panel.) The Loader automatically starts the loaded program if a transfer address was specified.

If more programs are to be loaded, place the next Paper Tape in the reader or set the Display Panel for Magnetic Tape or Cassette skipping, and depress RUN. This procedure starts the Loader executing at ORG +36, which is the continue location. The continue operation uses the current value of bias rounded up to a double word boundry, and does not reset it to X'A00'. Multiple relocatable programs are thus loaded one after another into adjacent areas of memory.

If it is desired to load a relocatable program other than at X'A00' in memory, it is necessary to redefine the bias value. To adjust the bias pointer, use the following procedure.

1. Change the fullword at ORG + 8 in the Loader to the desired bias value. The value contained in the fullword at ORG + 8 is rounded up to a double word boundry before loading begins.
2. Start the Loader executing at ORG + 6, rather than the normal start or continue location.

Note that the value of ORG + 8 remains until changed to a new value. The Loader can always be restarted at ORG + 6 which resets the bias to the value contained in ORG + 8.

Forward Reference Definitions

Load Modules output by the OS Library Loader involve forward references to symbols which are defined later in the program. The Loader uses a chaining procedure for satisfying any forward references at the time the symbol definition is encountered.

3.2 Operation

The steps required to load and operate the REL Loader are:

1. Manually enter the 50 Sequence into memory if it is not already there. Specify the device to be used for loading at location X'78', the Binary Input Device definition. See Appendix 3 for a listing of the 50 Sequence.

2. Place the REL Loader Tape or Cassette in the input device. Magnetic Tape and Cassette should be in the 'LOAD' position (LOAD POINT).

3. Initialize the Processor.

4. Execute at Location X'50'.

5. If an ASR 33 Teletype is being used as the input device, it is necessary to toggle the reader switch to START, which starts the tape moving. If an ASR 35 Teletype is in use, the Mode switch should be put in the KT position and the reader switch should be put in the RUN position to start the tape. If a High Speed Paper Tape Reader, Magnetic Tape, or Intertape Cassette is in use, the tape starts under program control.

6. If no input errors occur, the entire tape is read to the end, at which time the Processor halts with the Wait light ON, and 0000 contained in the Display Register.

7. Ready the program to be loaded on the input device. If the program to be loaded is relocatable and a specific bias value is required, enter the bias value into ORG + 8 and set the starting address to ORG + 6. If the program is absolute or if the current bias value is satisfactory, set the starting address to ORG and Depress RUN. If the program to be loaded is on Magnetic Tape or Cassette, place the number of file marks (hexadecimal) to be skipped into the low order byte of the data register of the Display Panel before depressing RUN. After the specified file marks have been skipped, the loader begins loading immediately without halting. If the user does not want to skip file marks, and loading is being done from Magnetic Tape or Cassette, the data register on the display panel must be cleared by depressing DTA, before RUN is depressed.

8. If improper control items are detected during the load, the tape stops, and the Processor halts with the Wait light ON and FFNN contained in the Display Register, where NN is the bad control item. When this occurs, it must be determined if the right loader is being used. That is, if the Object Tape involves EXTRNs or Common blocks, the OS Library Loader should be used. If appropriate, depress RUN to skip the improper data and proceed with the load.

9. If checksum errors are detected during the load, the tape stops and the Processor halts with the Wait light ON and 0001 contained in the Display Register. Reposition the tape to the start of the last record read, and depress RUN to reread the record. If parity errors occur with Magnetic Tape or Cassette, the record is automatically backspaced for re-trying.

10. When the load is complete, the tape stops. If no transfer address is specified on the tape, the Processor halts with the Wait light ON and 0000 contained in the Display Register. If a transfer address is specified, the Relocating Loader transfers directly to the location specified.

11. If more programs are to be loaded, return to Step 7 and repeat the process. This loading process is summarized in Figure 1.

3.3 REL Loader Bootstrapping

The Relocating Loader is provided in a Relocating Bootstrap form. When the tape is loaded using the 50 Sequence, the following events take place:

1. The 50 Sequence reads the Loader into memory.

2. The top of memory is then determined with a non-destructive search and the REL Loader is relocated a fixed distance (X'500') from the top of core or X'40000' whichever is less.

3. The Processor halts at ORG with the Wait light ON.

This sequence requires that the proper 50 Sequence be used, including the Binary Input Device Definition in X'78'. The 50 Sequence is shown in Appendix 3. When loading the REL Loader, memory from X'80'-X'CF' and X'1000'-X'1400' is used. Any programs in this area of memory are overwritten.

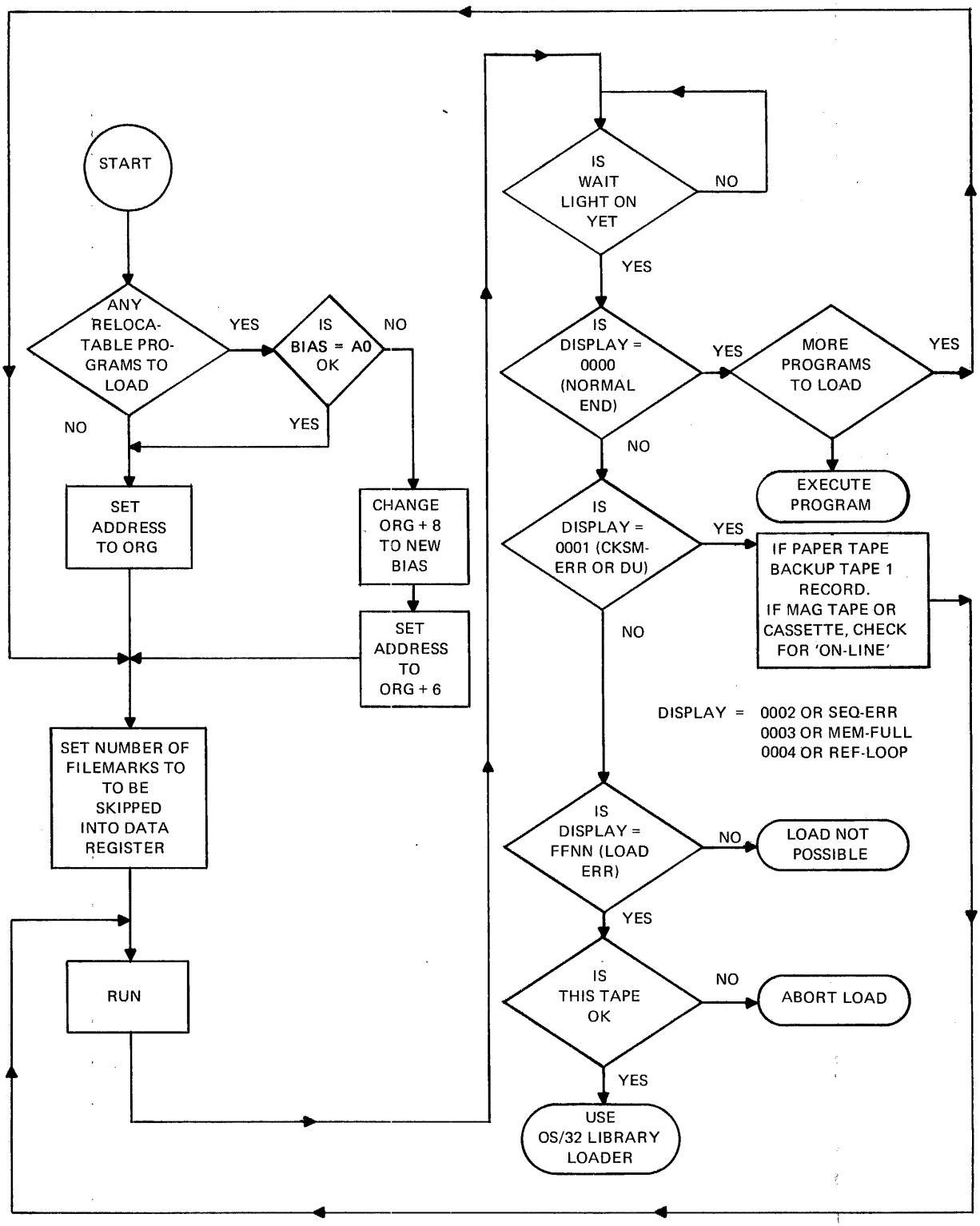


Figure 1. 32-Bit REL Loader Operating Procedures

4. THE OS/32 LIBRARY LOADER

This loader functions under the control of OS/32-ST or OS/32-MT, interactively accepting commands and responding with messages to the console operator. Through available operator commands, an Object Program Library File can be created on a bulk storage device. This file may be searched for a particular program, selectively copied onto another I/O unit, or added to under the operators control. Automatic link editing is available allowing the operator, with one command, to load all library programs required for any one particular job. A group of programs may be pseudo-loaded producing a single absolute load module which may be origined at any address, and then loaded by OS/32-ST Resident Loader or the 32-Bit Relocating Loader. This module contains forward references but no ENTRY or EXTRN symbols. A program loaded by the Library Loader may reference external symbols, define common blocks, and contain forward references.

4.1 Program Loading Conventions

Program segmentation is a means for allowing certain blocks of executable code to be shared when operating under OS/32-MT. Segmentation is accomplished by dividing a program into Pure and Impure segments at assembly time. The Pure segment is a sharable segment while the Impure segment is not. The OS/32 Library Loader is capable of loading programs under OS/32-ST that contain Pure segments, Impure segments, or both even though OS/32-ST does not support program segmentation. Certain ground rules have been formed to handle all possible cases. A program, such as CAL, that contains both Pure and Impure segments, may be loaded and executed by specifying a single bias value. The BI (Impure Bias) command is used to specify the bias for this type of program load operation. As another example, a FORTRAN program may be loaded with a single bias value (BI) even though the main program is Impure and the Run Time Library routines are Pure. All programs, Pure or Impure, are loaded contiguously when only this bias is specified. If the user desires to separate the Pure and Impure segments of a program or number of programs, he may specify a PURE bias using the PB command. In this case, pure segments are originated at the pure bias and impure segments at the impure bias. When the Pure bias (PB) is zero or unspecified, all segments are loaded at the BI value. The BI command automatically zeros the Pure bias value; therefore, when specifying both biases, the PB command should be entered after the BI command. The following table describes all cases of a program loading.

BIAS (BI)	PURE BIAS (PB)	PROGRAM CONTAINS		RESULT
		PURE SEGMENT	IMPURE SEGMENT	
0 (or unspecified)	0 (or unspecified)	Yes	No	Program loaded at PBOT (1st Location after loader)
0	0	No	Yes	Program loaded at PBOT (1st Location after loader)
0	0	Yes	Yes	Pure segment loaded at PBOT. Impure segment immediately follows pure segment.
B	0	Yes	No	Program loaded at B
B	0	No	Yes	Program loaded at B
B	0	Yes	Yes	Pure segment loaded at B. Impure segment immediately follows pure segment.
0	P	Yes	No	Program loaded at P
0	P	No	Yes	Program loaded at PBOT
0	P	Yes	Yes	Pure segment loaded at P. Impure segment loaded at PBOT
B	P	Yes	No	Program loaded at P
B	P	No	Yes	Program loaded at B
B	P	Yes	Yes	Pure segment loaded at P. Impure segment loaded at B

4.2 OS/32 Library Loader Operation and Commands

The Library Loader may be loaded by the OS/32-ST Resident Loader at the bottom of User Memory. Any I/O devices to be used by the loader should be assigned prior to starting the Loader at its origin. The Loader outputs the message "LIBLDR-R02" after an XOUT command indicating that it is ready to accept operator commands (which it requests from Logical Unit 5). Operator commands consist of a word (of which only the first two characters are decoded), followed optionally by a space and a hexadecimal or decimal operand followed optionally by another space and an 8 character program name. A carriage return terminates the command if it is entered via a Teletype. Optional fields which are not used are ignored. An example and description of each command follows. Arguments which specify logical units (LU) are entered in decimal. All other numeric arguments are in hexadecimal.

BIAS bbbbbb

This command sets a Loader Bias, i.e., the origin of the next relocatable program. This bias becomes the bias of the impure relocatable segment if a pure bias is subsequently specified by the PB command. bbbbbb is a hexadecimal operand from 1 to 6 hexadecimal digits. Immediately after loading any program, the bias value is set to the next available location above the highest address used. When the Loader is initially loaded, bias is set to the next location above the Loader. This value is not initialized when the Loader is restarted. If no operand is specified in this command, the bias is set to the next location above the loader. The BIAS command resets the PBIAS value to zero. The BIAS command may not be used during the output on an overlay module.

PBIAS bbbbbb

This command sets a loader bias for the origin of pure relocatable segments. After loading a program, unless a new PBIAS command is issued, the pure bias value is set to the next available location above the highest pure address used. When the loader is initially loaded, PBIAS is zero. This value is not initialized when the loader is restarted, but it is reset to zero by the BIAS and OUT commands. The PBIAS command may not be used during the output on an overlay module.

FIND LU xxxxxxxx

This command causes logical unit LU to be searched until a program labeled xxxxxxxx is found. The unit is then backspaced to the beginning of the found program. The program may now be Loaded or Linked. The logical unit is not rewound by the OS/32 Library Loader R01 when this FIND LU command is given. This command should be used to position a file containing several programs. If the file is not on a rewindable device, it should be backspaced one record after this command is completed before attempting to load the found program. In the event that the requested program is not found, the FIND command terminates whenever an end-of-file or device end is encountered on the device. In that case, the messages

```
EOF
xxxxxxxx NOT FOUND
```

are logged on the console device.

LOAD LU xxxxxxxx

This command is used to initialize the Loader and load a program at the current bias values. Before the program is loaded, the Loader's symbol table is cleared, and previously defined common blocks are released. This command should be used to load the first of a group of user programs not related to any previous job. If a program label is specified in the command, logical unit LU is first searched for program xxxxxxxx as described above under FIND. If no Label is specified, the first program encountered on logical unit LU is loaded. Following a LOAD operation, the logical unit specified is left positioned past the end of the program loaded.

LINK LU xxxxxxxx

This command is used to load additional programs needed after using the LOAD command to load the first program. The symbol table is not cleared, so the loaded program may be linked to any previously loaded program, and may reference previously defined common blocks. The program is loaded using the current bias values. If a program Label is specified in the command, logical unit LU is first searched for program xxxxxxxx as described above under FIND. If no Label is specified, the first program encountered on logical unit LU is loaded and linked. Following a LINK operation, the logical unit specified is left positioned past the end of the program loaded.

LINKFILE LU XXXXXXXX

This command provides an internal iteration of the LINK command. It link-loads all programs and subroutines in a file up to an ENDVOL or file mark. The name field is optional. If not included, linking will begin from files current position. If a program name is included in the command, the input file will be positioned to that program label (see FIND command) before linking begins.

EDIT LU xxxxxxxx

This command is used to search a library tape or file, and selectively load those programs needed to satisfy external references (EXTRNs) within programs previously loaded. The criterion for loading during an EDIT operation is a match between the Label of a program in the library file and the symbolic name of any EXTRN in the Loader's symbol table. The EDIT command causes the following to occur.

1. If a program Label is specified in the command, logical unit LU is searched for program xxxxxxxx as described above under FIND before starting the EDIT process. If no label is specified, the logical unit is processed from its current position.

2. The Loader's symbol table is searched for undefined EXTRNs. If there are none, the operation terminates.

3. If there are undefined EXTRNs in the symbol table, logical unit LU is searched forward from its current position for a Label matching an undefined EXTRN. If such a Label is found, that program is loaded and linked, and Step 2 is repeated. If an end-of-file or device end is encountered while searching, an appropriate message is logged and the operation terminates. If the Label ENDVOL is read, the EDIT operation is terminated. This Label may be placed at the end of each tape of a Paper Tape program library to prevent reading off the end of the tape.

Following an EDIT operation, the MAP command can be used to determine if any more undefined EXTRNs remain in the symbol table.

MAP LU

This command outputs a Memory Map to logical unit LU. This may be a map showing the location of programs loaded into memory, or it may show the locations of programs output in a load module. In either case, the map includes only those programs processed during and after the last LOAD operation. (LINK and EDIT operations add to the map, LOAD begins a new one.) The map shows the starting address of each labeled program, the next available address, the location of every ENTRY defined, the starting address of each common block defined, and a list of any undefined EXTRNs. An example of a memory map appears in Figure 2. Under PROGRAMS in the map, the last value shown which has no name is the next available address in memory, which is the current value of BIAS. Under COMMON-BLOCKS in the map, Blank Common is identified by the Label / / .

```
PROGRAMS:

NAME        IMPURE    PURE    ABS       NAME        IMPURE    PURE    ABS
CLEANUP     00230A            000030    CMDP.M03    007238


ENTRY POINTS:

002118   SUB1                 00236E   .U           00238E   .V

COMMON-BLOCKS:

002010   R                    002018   X            00201C   Y       002020   W

002000   //

UNDEFINED:

ZERO                          @I                    .Q                .P
```

NOTE

Where no numbers appear in this portion of the map, no code of that type was encountered when loading.

Figure 2. Memory Map (As listed by the OS/32 Library Loader)

If an overlay structure is being built, the map always reflects the root segment but only the last overlay segment appears. Upon entering the overlay command (OV), all entries to programs above the overlay point are removed.

AMAP LU

This command, like the MAP command, outputs a memory map to logical unit LU. Symbols are listed in alphabetical rather than memory address order.

BC nnnnnn

LC nnnnnn

These commands must be issued before the LOAD command if the program being loaded allocates Common. The Loader handles two types of Common. They are Labeled Common, and Blank Common. The chief difference between the two is that Labeled Common may be preloaded with data (FORTRAN BLOCK DATA) while Blank Common may not. The Loader allocates memory for common variables and links common variable references with associated blocks of memory at load time. In order to use memory efficiently, the Loader must know in advance the total length, in bytes, of each kind of Common used. BC sets the maximum length of Blank Common and LC sets the maximum length of Labeled Common to nnnnnn, where nnnnnn is the number of bytes to be reserved for common, expressed in hexadecimal. When Common is declared to the Library Loader by the BC or LC command, the Loader places the common area beginning at the bias value. The program, as a result, is actually loaded at BIAS + BC + LC.

GO

This command transfers control from the OS/32 Library Loader to the transfer address of the loaded programs. If no transfer address is specified, the message CMD-ERR is logged.

LABEL LU xxxxxxxx

This command is used in conjunction with adding programs to a library file. If the program to be added to the library does not have a label, then this command can be used to give a program a label in the file. The command causes the loader to output one record to logical unit LU containing the program label xxxxxxxx. The label should not exceed eight alphanumeric digits. If it does, only the first eight are used. This command could be used prior to an assembly or compilation which writes the binary object program to logical unit LU.

This command should never be used between an OUT and XOUT command. If it is, the command is rejected, and the message CMD-ERR is logged.

OUT LU xxxxxxxx

This command selects the output mode which is used to generate a load module rather than load programs into memory. In this mode, during LOAD, LINK and EDIT operations, all data that would normally be loaded into memory is output in loader format to logical unit LU. All programs are output as absolute code. The OUT command sets the bias value to the first location above the current Operating System. The BIAS command must be issued after the OUT command if any other bias value is required. The pure bias value is reset to zero. Any external references or common references are resolved and converted to forward references without symbolic names. Thus, a load module program can be loaded by the 32-bit REL Loader or the OS/32-ST Resident Loader. By using the default value for the bias, and loading the load module with the OS/32-ST Resident Loader, a user can effectively overlay the Library Loader at run time, although its features may be used at load time.

If a program Label is specified in the command, a label record with the name xxxxxxxx is generated as described with the LABEL command prior to selecting the output mode.

XOUT

This command is used in conjunction with generating a load module. This command, which has no argument, generates the final record of the load module, and cancels the output mode. A typical command sequence to create a load module would be OUT, BIAS, LOAD, LINK, EDIT and XOUT.

COPY LU, LU xxxxxxxx

This command causes one program to be copied from the first logical unit to the second. If a program label is specified in the command, the first logical unit is first searched for program xxxxxxxx as described under FIND. If no Label is specified, the first program encountered on the first logical unit is copied. Following a COPY operation, the first logical unit specified is left positioned past the end of the program copied.

WF LU

This command writes an end of file to logical unit LU.

DUPE LU, LU xxxxxxxx

This command causes programs to be copied continuously from the first logical unit to the second until program xxxxxxxx is found. The specified program is not copied and the operation terminates. If no Label is specified, programs are copied until an end of file or device end on the first logical unit is detected.

To duplicate an entire file of programs from Logical Unit 1 to Logical Unit 2, the following command sequence is sufficient.

```
RW   2
RW   1
DU   1,2
```

TABLE LU, LU

This command scans the first logical unit and lists on the second logical unit, all program labels found from the exact position of the file when this command was given. To table all program labels the user must issue a rewind command or know the file is at the beginning. Thus a table of contents of a file containing many labeled programs may be obtained. If the record containing the label of a program has been passed before the Table command is given, that program is not listed. Any program in the file without a label is not shown in the list.

END

This command is used to return control to the operating system at end of job. When the system pointers are repositioned before returning control to the operating system, the symbol table is lost. To exit temporarily from the Library Loader without losing the symbol table, the PAuse command should be used.

OV xxxxxx

The form of this Overlay command is OV xxxxxx, where xxxxxx = the sum of all active storage areas requested in the program via SVC 2 (Get Storage). If not specified, a default value of X'240' is assumed. This command is used to indicate that a subroutine about to be linked to a program is used as an overlay subroutine. The OV command, used prior to linking a subroutine and subsequent subroutines causes them to occupy a common area of memory (i.e., each subroutine is originated at the first available location above the main program). The OV command must be used with the OUT feature of the library loader to create overlay modules. These modules may be called in FORTRAN by subroutine IFETCH and in assembly language programs by SVC 5.

The OV command's operand specifies the area to be reserved between the main program and the first overlay. The operand is a number of bytes (hexadecimal) rounded up to a fullword value. The default value (X'240' bytes) is the amount of storage requested by an executing FORTRAN program; i.e., the operand need not be specified for FORTRAN overlay operations. For assembly language programs, the operand should specify the sum of the active storage areas requested or can be used to delete the reserved area (EX: OV 0).

Ex: If, in a program, the following storage requests were made:

```
GET X'100'
GET X'200'
RELEASE X'200'
RELEASE X'100'
```

then the OV argument is 100 + 200 = 300

If requests are:

```
GET X'100'
RELEASE X'100'
GET X'50'
RELEASE X'50'
```

then the OV argument is 100.

As an example of overlay establishing procedures, assume a FORTRAN application program and two subroutines which are to be linked as overlays. The program and its subroutines have been compiled and their object programs are on Paper Tape. The main program calls the first overlay via Logical Unit 2 and the second via Logical Unit 3 (established by the FORTRAN "CALL" statement).

In the following example, Logical Unit 4 is assigned to the Paper Tape Reader, Logical Unit 6 is assigned to the FORTRAN Run-Time Library device, and Logical Units 1, 2, and 3 are assigned to the output devices which will contain the main program, Overlay 1, and Overlay 2, modules respectively.

```
OUT   1          Set the OUT option for Logical Unit 1.
LO    4          Load the main programs.
ED    6          Edit with the Run Time Library.
XOUT
WF    1          Write file marks on the output files.
OV               Indicate an overlay is to be linked.
OUT   2          Set the OUT option for Logical Unit 2.
LI    4          Link First Overlay.
ED    6          Edit with the Run Time Library.
XOUT
WF    2
OV               Indicate another overlay is to be linked.
OUT   3          Set the OUT option for Logical Unit 3.
LI    4          Link Second overlay.
ED    6          Edit with the Run Time Library.
XOUT             Terminate OUT option.
WF    3
EN               Terminate Library Loader.
```

The first OV command following a LOAD command establishes the BIAS location for all overlays to follow.

The commands must be entered in the correct sequence, i.e.,

```
OV
OUT
LI and/or ED
XO
OV
OUT
LI and/or ED
XO
      etc.
```

The OV command cannot be entered while the Library Loader is in "out mode", nor while a previous overlay is still being built. The commands BIAS and PBIAS may not be entered during the output of an overlay, i.e., between the OV and XO commands.

LG x, LU

The LG command has the form x, LU where LU is the logical unit of a list device. All loader commands read from the loader command input device (Logical Unit 5) are logged on logical unit LU. If LU = 0, no logging occurs. x may be a digit of 0 or 1 and is used to suppress the Prompt message after each command in cases where commands are being read from a batch input device or Logical Unit 5.

Examples:

```
LG  0,4          Log on Unit 4
LG  1,3          Log on Unit 3, suppress Prompts.
LG  0,3          Continue logging with Prompts.
LG  1,0          Stop logging, suppress Prompts.
LG  0,0          No logging, reinstate Prompts.
```

By default, logging is suppressed, and the 'LIBLDR' message is only output if commands are issued interactively.

PAUSE

This command causes the loader to issue an SVC 2 PAUSE. The loader may subsequently be restarted by the OS CONTINUE command. Upon loader continuation, the status of the loader is the same as prior to the pause. Do not use the PAUSE command to assign additional disc files. Use the EN command to release memory and restart the Library Loader once all assignments have been made.

TOP xxxxxx

This command sets the highest memory address available to the loader for loading user programs. Address xxxxxx is a hexadecimal address. The TOP command overrides the loader assumption that all memory on the system is available. The TOP command may be used to build load modules for execution on systems with more or less memory than the system used to build the load module. This command deletes the symbol table.

The OS/32 Library Loader attempts to make use of all available memory under OS/32-ST for the loader symbol table. To insure that enough memory is available for load time direct access file use, the user should close all unneeded files and open all files required by the loader before entering the loader.

RW LU

This command rewinds logical unit LU.

FF LU

This command forward spaces the LU to a file mark and positions the LU past the file mark. If no file mark is found a printout of I/O Device Error occurs.

BF LU

This command backspaces the LU to before the file mark. If no file mark is found a printout of I/O Device Error occurs.

FR LU

This command moves the LU forward to the next record on the file.

BR LU

This command backspaces the LU one record on the file.

4.3 Library Loader Messages

MEM-FULL

This message is logged if a program being loaded exceeds the available memory. This may happen if too little Common space is allocated (if the program defined Common) or if too large a program is loaded. A table of Labels, Common, and external symbol names is kept, building downward from the top of available memory (see memory maps). If the program overlaps the table, the above message is logged and the load is aborted.

CMD-ERR

This message is logged when the loader does not recognize an operator command, if a LABEL command is found between an OUT and XOUT command, if GO is issued when no transfer address has been specified in any loaded program, if a LINK or EDIT command is issued before any program has been loaded, or if commands used to build overlays are given in an illegal sequence.

* * NO PROGRAMS * *

This message is logged if a MAP is requested and no labeled programs have been loaded.

EOF

A file mark or ENDVOL has been read during binary input.

EOM

An END OF MEDIA has been encountered during a binary input.

CKSM ERR

A record has been read in which the checksum does not match the checksum computed by the loader. Following this message, the loader pauses. If the input device is Paper Tape, it may be manually backspaced one record before continuing the load. Bulk storage devices are automatically backspaced on CONTINUE.

SEQ ERR

A record has been read out of sequence. Reposition the input unit and CONTINUE.

REF-LOOP

A Loop has been found in a forward or external reference thread. The object program has been generated incorrectly. This message is also output if the XO command is given during the building of overlay modules when unresolved external references remain in the completed overlay.

M xxxxxxxx

An entry point (xxxxxxxx) has been defined more than once. The load continues and the previous definition remains in effect. The new definition is ignored.

LOAD ABORTED

An unrecoverable error condition has caused a load in progress to be aborted. It is necessary to repeat the original LOAD command, as the table may contain incorrect symbol locations.

ADRS-ERR

A program has attempted to load below the top of the Loader. The program may contain absolute data at such an address, or the BIAS may be set incorrectly. The load process is aborted. This is also output if a program with references or definitions of halfword address constants is loaded above 64K, or if a subsequent declaration of a Labelled Common Block is greater than the length of the first declaration of that Common Block.

DEV END

A device went Off-Line during input.

I/O DEV ERR xxxx

A parity error has been detected on input or an Off-Line condition has been detected on output. The OS status byte and device address in hexadecimal-notation appear.

LOAD ERR

An illegal control item was detected during a load.

xxxxxxxx NOT FOUND

An end of device condition has occurred before locating the label xxxxxxxx during a search operation.

If commands are being issued in batch mode with logging suppressed, and an error is found, then the last command obeyed will be output to the console device.

If the loading process is aborted in batch mode, the loader returns control to the Operating System via SVC 3,1, indicating that it was unsuccessful. If commands are being issued interactively then the loader reads the next command from the console device.

4.4 Library Creation

To create a program library from Paper Tape to a Bulk Storage device such as Cassette, Magnetic Tape, Disc, or Drum, use the following procedure. Assume the Paper Tape Reader has been assigned to Logical Unit 1 and a Magnetic Tape unit has been assigned to Logical Unit 2.

1. Load the OS/32 Library Loader Program and execute.
2. Rewind the Magnetic Tape.
3. Copy the OS/32 Library Loader Paper Tape to the Magnetic Tape.

```
CO   1,2
```

4. Copy any other desired programs such as the CAL ASSEMBLER, the FORTRAN V Compiler, or any user written program.

```
CO   1,2
```

5 Copy the RUN TIME LIBRARY Paper Tape to Magnetic Tape with the command.

```
DU   1,2 ENDVOL
```

This command should be repeated once for each separate Run Time Library tape.

6. Other programs may be added on to the end of the library at a future time with the CO or DU commands.

NOTE

User written subroutines may be edited from the library if the following conditions are met:

1. The subroutine contains a PROG statement, or a label is created with the LABEL command.
2. The subroutine contains an ENTRY statement with a name that is the same as in the label.
3. The user's main program contains an EXTRN statement which refers to the name of the subroutine in the library.

4.5 Interface between the Library Loader and OS/32-ST.

Memory management in OS/32-ST involves the setting and maintaining of system parameters:

User bottom	-- UBOT
User top	-- UTOP
Memory top	-- CTOP
File Control Block Bottom	-- FBOT

When the Library Loader is loaded, UBOT remains immediately above the operating system, and UTOP and CTOP are placed immediately above the Library Loader.

When devices and/or files are assigned, File Control Blocks are set up at the top of memory, and FBOT is moved down accordingly from CTOP.

When the Library Loader is entered, it moves CTOP and UTOP as high as possible without over-running the File Control Block. Programs may be loaded between the top of the Library Loader and the top of available memory, CTOP.

The Loader may be PAused to examine or modify memory, display system parameters, etc., but no further devices or files may be assigned because there is no room between FBOT and CTOP for more File Control Blocks.

During the loading and linking process, the Loader maintains a pointer to the highest address loaded. When the EN or GO command is received, UTOP is repositioned on a 256-byte boundary above UTOP.

5. OS/32 DIRECT ACCESS BOOTSTRAP LOADER DESCRIPTION

This section describes the operating procedure required to load an OS/32 MT or OS/32 ST system with the OS/32 Direct Access Bootstrap Loader (Program Number 03-074R02). The system to be loaded must reside in memory image format on a direct access device. The direct access devices supported by this program are the 2.5, 10, 40, 66, and 250 Megabyte Disc Systems.

This section also contains a description of the internal logic of the program.

5.1 Requirements

Minimum hardware required to execute this loader:

- INTERDATA Series 32 Processor
- Direct access I/O device; 2.5, 10, 40, 66, or 250 Megabyte Disc
- A binary input device from which to read the loader

5.2 OS Selector

The OS/32 Direct Access Bootstrap Loader assumes that the specified volume has a file directory containing file descriptors of the form:

OS32xxxx.hhh

where: xxxx = any ASCII characters
hhh = Extension field (X'000' - X'FFF')

In order to load a particular OS Image, the user must copy the corresponding extension field (hhh) into the Memory Locations X'7E' - X'7F' as an OS Selector Number.

The loader searches the file directory for a contiguous file with a filename starting with the characters "OS32" and an extension field which matches the specified OS Selector Number. When it finds this file, it loads it into memory. If no such file is found, the loader terminates with an error code of 1 in the Display Panel.

NOTES

1. If the volume is initialized by using either the OS/32 ST INITIALIZE command or Release 00 of the OS Disc Initialize Utility (Program Number 03-081), it may contain an unnamed OS image. In this case, the volume descriptor (Sector 0, Track 0, Cylinder 0) has a pointer to this image. It also contains the start and end address of the image.

   In order to load this image, an OS Selector Number of 000 must be specified.

2. It is possible to have an unnamed OS image (as explained in Note 1 above) and also a filename with an extension of 000 (e.g., OS32xxx.000) on the same volume. In this case, an OS Selector Number of 000 causes the loader to load the OS specified by the filename in the directory.

3. An OS Image file must be a contiguous file. Files other than contiguous files are ignored by the loader.

4. If there are two (2) images on the volume with the same extension (hhh), the loader loads the first one in the directory.

5.3 Operating Procedures

The OS/32 Direct Access Bootstrap Loader is a self loading tape. The '50' Sequence loads a 'boot' section (Segment 1) which loads the remainder of the tape at Location X'1000' and executes it.

The '50' Sequence should be entered from the Display Panel as follows:

Location	Contents
50	'D500'
52	'00CF'
54	'4300'
56	'0080'

For the loader to function properly, memory cells X'78' through X'7F' must be set up properly. Refer to Table 2 for the possible values of these locations.

After these cells have been initialized, the user need only position the bootstrap loader tape in the binary input device and start the Processor executing instructions at X'50'.

The bootstrap loader begins by scanning the directory of the Direct Access device specified. It seeks a contiguous file whose name begins with the four alphanumerics 'OS32'. When it locates such a file it compares the file's extension to the OS Selector. If a match occurs, the file is loaded and executed. If no match, the directory scanning continues.

Any errors, either I/O or logical, cause a halt. The number shown on the display register indicates the error codes. Refer to Table 5 for the Error Code meanings.

TABLE 2. MEMORY INITIALIZATION CONTENTS

Memory Location	Contents
X'78', X'79'	Address and output command of binary input unit (see Table 3): Example X'1399'
X'7A', X'7B'	Address and Device Code of Direct Access Device (see Table 4): Example X'C631'
X'7C', X'7D'	Address of Controller and Selector Channel (see Table 4): Example X'B6F0'
X'7E', X'7F'	OS Selector Number (see Section 52 on OS Selector): Example X'0005'

TABLE 3. STANDARD BINARY INPUT DEVICE ADDRESSES AND OUTPUT COMMANDS

Device	Standard Address and Output Commands
TTY	X'0294'
Paper Tape Reader	X'1399'
Magnetic Tape	X'85A1'
Intertape Cassette	X'45A1' or X'55A1'

TABLE 4. STANDARD DIRECT ACCESS DEVICE ADDRESSES AND DEVICE CODES

Device	Standard Address	Device Code	Controller Address	SELCH Address
2.5MB Disc	X'C6'	X'31'	X'B6'	X'F0'
10MB Disc (removable platter)	X'C6'	X'33'	X'B6'	X'F0'
10MB Disc (fixed platter)	X'C7'	X'32'	X'B6'	X'F0'
40MB Disc	X'FC'	X'34'	X'FB'	X'F0'
66MB Disc	X'FC'	X'35'	X'FB'	X'F0'
250MB Disc	X'FC'	X'36'	X'FB'	X'F0'

TABLE 5. ERROR CODES

Error Code	Explanation
1	1. No OS Image Found; or 2. OS Image with the File Name Extension same as the OS Selector Number (hhh) not found.
2	Status Error for a 2.5 or 10MB disc (device unavailable)
3	Status Error for a 2.5 or 10MB Disc (seek incomplete, not ready)
4	Status Error for a 40, 66, or 250MB disc (unsafe, not ready, device unavailable)
5	Status Error for a 40, 66, or 250MB disc (unsafe, seek incomplete, or not ready)
6	Data Transfer Error (e.g., data parity error)
7	OS Image found larger than available memory

APPENDIX 1

SUMMARY OF OS LIBRARY LOADER COMMANDS

CMD	ARG1	ARG2	MEANING
BIAS	BBBBBB		Set Bias to BBBBBB
BIAS			Set Bias to End of Loader
PBIAS	PPPPPP		Set Pure Bias to PPPPPP
BC	LLLLLL		Set Length of Blank Common
LC	LLLLLL		Set Length of Labeled Common
TOP	LLLLLL		Set Top available address to LLLLLL
OUT	LU	LABEL	Output Labeled Load-Module to LU
OUT	LU		Output Load Module to LU
XOUT			End Load Module, write last record
GO			Transfer to Loaded Program
MAP	LU		Output Memory Map to LU
LABEL	LU	NAME	Write Label Record on File LU
FIND	LU	NAME	Position LU to Start of Named Program
COPY	LU, LU		Copy one program from LU A to LU B
COPY	LU, LU	NAME	Find and Copy Named Program
DUPE	LU, LU		Duplicate until EOF or Dev END
DUPE	LU, LU	NAME	Duplicate until Name is read
LOAD	LU		Purge Table, load one program from LU
LOAD	LU	NAME	Find and Load a program
LINK	LU		Load one program from LU, Linking
EDIT	LU		Load only referenced programs from LU
EDIT	LU	NAME	Find Named Label and Edit
RW	LU		Rewind LU
WF	LU		Write a File-Mark on LU
TABLE	LU, LU		Write Table of Contents to LU
OVLY	XXXXXX		Set Get Storage Space to XXXXXX
PAUSE			Pause Loader
END			Exit to the OS
LG	X, LU		Set command log option to LU
AMAP(AM)	LU		Out Memory Map sorted to LU
FF	LU		Forward-Space-File-Mark
BF	LU		Back-Space-File-Mark
FR	LU		Forward-Space-One Record
BR	LU		Back-Space-One Record
LINK	LU	NAME	Link program NAME
LINKFILE	LU		Link file of programs
LINKFILE	LU	NAME	Link file of programs from NAME

APPENDIX 2

SUMMARY OF 32-BIT RELOCATING LOADER OPERATION

All values below are expressed in hexadecimal:

Starting address after Boot Load*	nB00
Restart address	nB00
BIAS define start address	nB06
BIAS definition value	nB08
Continue address	nB1E
Approximate Loader size	X'500'
Illegal Control items	3, 5, 7, 9, B, C, E, 10, 15, 17
Ignored control items	D, F, 18

* n = 7, F, 17, 1F. . . for memory sizes 32K, 64K, 96K, 128K . . . etc.

The following Display indications are appropriate to the REL LOADER.

Display Lights	Condition	Comment
0000	Normal End	Load Complete
0001	Checksum Error	Following checksum or sequence number error, re-position paper tape and depress RUN to re-read. If magnetic tape or cassette - parity error causes 5 retries before returning this status.
0002	Sequence Error	
0003	Attempt to load over loader	
0004	Ref/Def Chain Loop	
FFnn	Load Error	Improper control item detected where nn is the bad item. Depress RUN to ignore the data and continue.
		Refer to Section 2 for a definition of loader control item.

APPENDIX 3

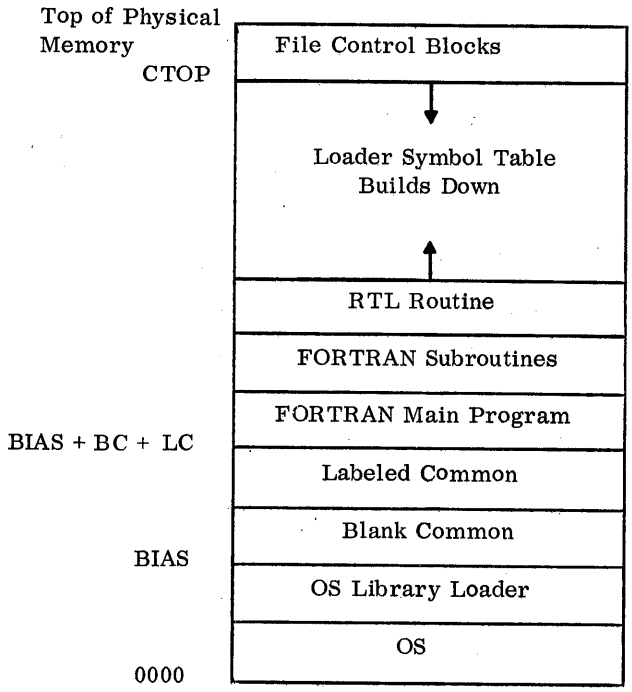
50 SEQUENCE LOADER (EXECUTE AT LOCATION X'50')

```
X'0050'     D500 X'CF'      AL      X'CF'
            4300 0080       B       X'80'
X'0078'     0294    BINDV   DC      X'0294'
```

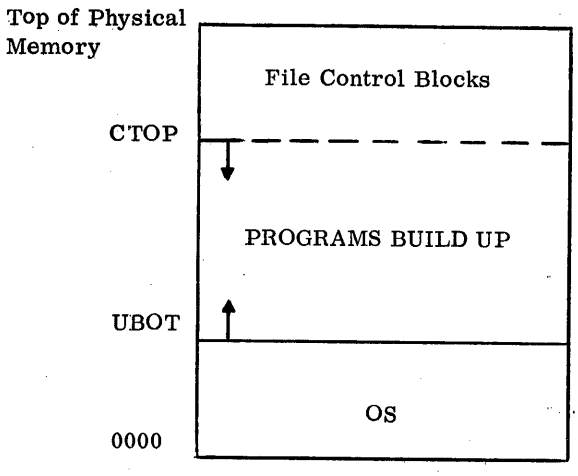
APPENDIX 4

LOADER MEMORY MAPS

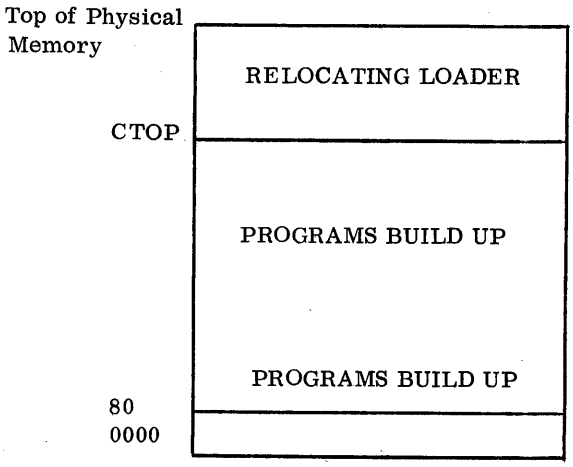
OS/32 LIBRARY LOADER



OS/32-ST RESIDENT LOADER

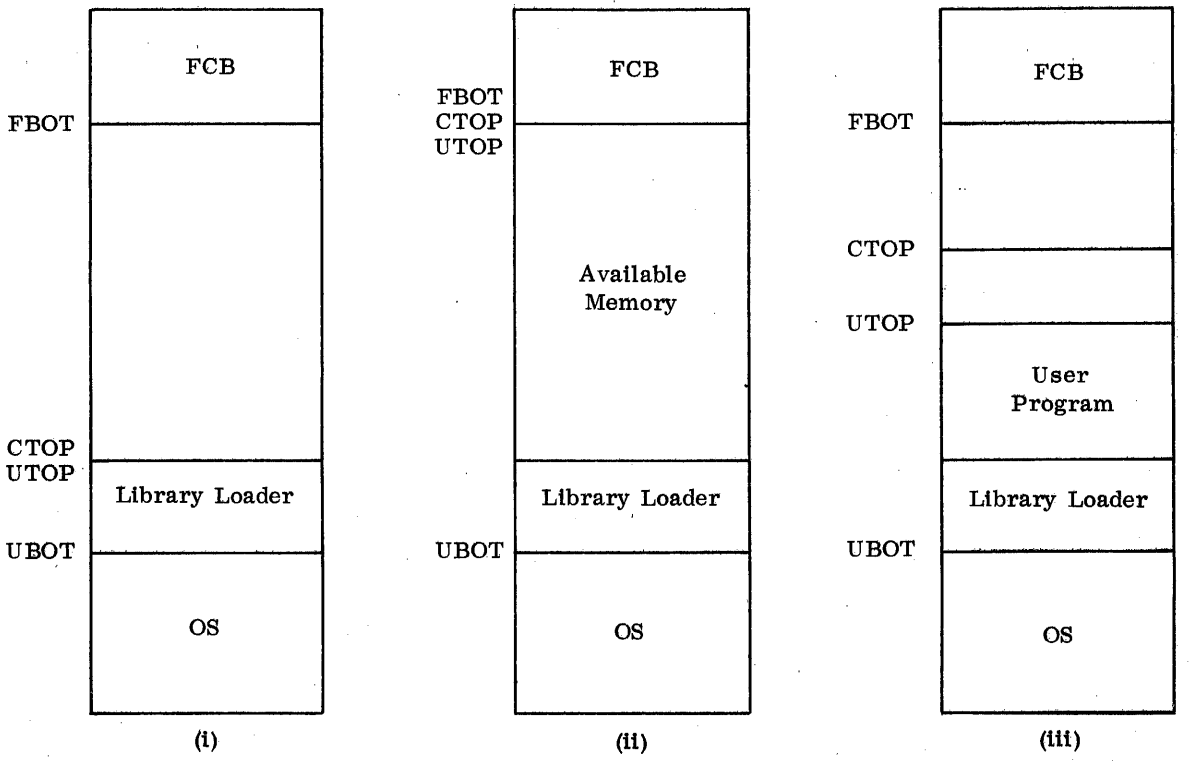


32-BIT RELOCATING LOADER



CTOP = TOP OF USER MEMORY

Library Loader Memory Maps - Showing CTOP, UTOP, and UBOT



APPENDIX 5

USE OF THE BOOT PUNCHER

1. DESCRIPTION

The OS BOOT PUNCHER (03-070) converts the object versions of the 32-bit Relocating Loader (03-067) and the OS/32 Bulk Storage Boot Strap Loader (03-074) into 50-Sequence loadable form. These programs are provided in object form within the OS/32-MT Package (S90-006) and the OS/32-ST Package (S90-005).

Programs in 50-Sequence loadable form can not be copied using the standard copy utilities. The Boot Puncher creates 50-Sequence loadable copies from the object code.

2. SYSTEM REQUIREMENTS

The OS/32 Boot Puncher is designed to run on any INTERDATA 7/32 or 8/32, using either the OS/32-ST (03-075) or the OS/32-MT (03-071) operating system. It requires a minimum of 2.5K of memory above the OS.

3. OPERATIONS

3.1 Logical Unit Assignments

The OS/32 Boot Puncher uses the following logical units:

LU	FUNCTION
1	The input device from which the loader in 32-bit loader format is to be read.
2	The output device to which the loader is to be written.

The input device may be any physical device which supports 126 byte record object data. The output device may be any device which supports variable length records, such as a paper tape, magnetic tape or cassette, but it cannot be a disc file.

If logical unit one is a disc file with a record length other than 126 bytes, or if logical unit two is a disc file the following message is output.

ILLEGAL ASSIGNMENT ON LU X

where X denotes the logical unit where the error occured. The task will then go to end-of-task with a code of one.

3.2 Execution

The OS/32 Boot Puncher can be executed by either of two methods. The user has the option of either assigning both logical units manually, or he can pass the device names as parameters with the START command.

3.2.1 Manually Assigning Logical Units

A. Load OS/32 Boot Puncher
B. Assign logical unit one to the input device
C. Assign logical unit two to the output device
D. Start

3.2.2 Passing devices as Parameters

A. Load OS/32 Boot Puncher
B. ST ,IN,OUT

where IN is the file descriptor of the input device, and OUT is the file descriptor of the output device.

3.3 Termination

Before the OS/32 Boot Puncher goes to end-of-task, it closes both logical unit one and logical unit two. If there were no errors during execution, an end-of-task code of zero is returned. If there were errors during execution, an end-of-task code of one is returned.

4. ERROR MESSAGES

The following is a summary of all the OS/32 Boot Puncher error messages. All error messages are output to the system console and are of the form:

ASGN-ERR TYPE=XXXX LU=YY

where XXXX is the error type of I/O and file access
YY is the logical unit where error occurred.

The possible values of the TYPE=XXXX field are:

I/O:	ILLU	(illegal or unassigned LU)
	PRTY	(parity or recoverable errors)
	UNRV	(unrecoverable errors)
	EOF	(end of file)
	EOM	(end of medium)
	DU	(device unavailable)
	FUNC	(invalid function for device)
files:	FUNC	(illegal function)
	LU	(illegal LU)
	VOL	(no such volume/device)
	NAME	(mismatched FILENAME.EXT)
	SIZE	(erroneous record length or size)
	PROT	(mismatched protection key)
	PRIV	(mismatched access privilege)
	BUFF	(unable to obtain FCB)
	TYPE	(non-direct access device or off-line)

In addition to the above error messages, if either of the logical units is not assigned, the message

8100 ERROR:LU=XX

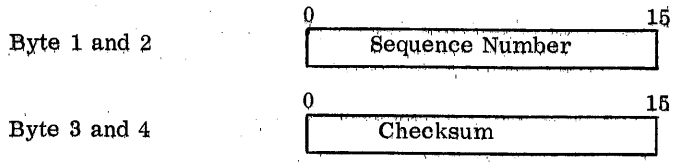
appears, where XX is the logical unit which was not assigned. The task is then paused and the user can then assign the logical unit and continue.

If any of the error messages above appears except for logical unit unassigned, the task goes to end-of-task with a code of one.

APPENDIX 6

INTERDATA 32-BIT LOADER FORMAT

Standard loader format binary object tapes are divided into records separated by inter-record gaps. Each record contains 126 bytes of information. The first four bytes of data are organized as follows:



The sequence numbers are sequential negative integers -1, -2, -3, etc., represented in two's complement form. The first record in a program must have sequence number -1. Subsequent records must be in proper order to be loaded.

The checksum is an Exclusive OR sum of all halfwords in the record, except itself, plus a halfword of all ONEs.

The remainder of the record is a sequence of items; an item is a byte of loader information. There are two types of items; control items and data items. Each control item is followed by a certain number (which might be zero) of data items. The control items and their meanings are listed in Table A6-1.

Physically, three tape formats are used, all of which contain 126 bytes of data per record with blank leader or inter-record gaps between records.

M11 Format: This format is used when punching Paper Tape on a High Speed Punch. The first non-blank character of each record contains an X'F0'. This character is followed by 126 frames, each containing on data byte. Thus, a complete record is 127 frames of which the first is ignored.

A variation of this format occurs when punching object programs on a Teletype punch. A special set of non-printing characters is used, such that only the four low order bits of each frame contain data. (See Table A6-2). If one of these special characters is encountered before an X'F0' character is read the record is assumed to be of this format. Because only 4 bits of each frame are used, the 126 byte record is Punched in 252 frames of Paper Tape, of which the upper 4 bits are ignored by the loader input routine. INTERDATA does not supply object programs in this form, but any object program punched on a Teletype by INTERDATA software is output in this 4 bit zoned format.

M21 and M31 Format: These formats are used with Magnetic Tape and Cassette (Intertape) devices. They are identical to the M11 format except that there is no X'F0' character. Additional parity checking is accomplished in the Magnetic Tape and Intertape hardware to insure a higher degree of data validity in addition to the checksum.

M51 Format: This format is used to describe object programs supplied on disc packs. It is identical to the M21 and M31 formats.

TABLE A6-1. CONTROL ITEM DEFINITION

Control Item	Meaning	Number of Data Items Following
0	End of Record	None
1	End of Program	None
2	Reset sequence number	None
3	Block data indicator	8 byte name, 3 byte displacement, any absolute data item (20-5B)
4	Absolute program address	3 byte address
5	Pure relocatable program address	3 byte address
6	Impure relocatable program address	3 byte address
7	2 bytes of pure rel data	2 byte address
8	2 bytes of impure rel data	2 byte address
9	4 bytes of pure rel data	4 byte address
A	4 bytes of impure rel data	4 byte address
B	Common reference	8 byte name, 3 byte displacement
C	EXTRN	8 byte name, followed by item 4, 5, or 6
D	ENTRY	8 byte name followed by item 4, 5, or 6
E	Common Definition	8 byte name followed by a 3 byte length
F	Program Label	8 character name
10	3 bytes abs and 3 bytes pure rel	6 bytes
11	3 bytes abs and 3 bytes impure rel	6 bytes
12	Load program transfer address	Item 4, 5, or 6
13	Define start of chain (reference)	Item 4, 5, or 6
14	Load chain definition address	Item 4, 5, or 6
15	2 bytes abs and 2 bytes pure rel	4 bytes
16	2 bytes abs and 2 bytes impure rel	4 bytes
17	Short Form EXTRN	8 byte name and Item 4, 5, or 6
18	Length of Impure and Pure segments	3 byte impure length and 3 byte pure length
19	Perform fullword chain	None
1A	Perform halfword chain	None
1B	No Operation	None
1C	2-byte pure translation table address	2 bytes
1D	2-byte impure translation table address	2 bytes
20	2 bytes absolute data	2 bytes
21	4 bytes absolute data	4 bytes
22	6 bytes absolute data	6 bytes
23	8 bytes absolute data	8 bytes
.	.	.
.	.	.
.	.	.
5B	120 bytes absolute data	120 bytes

TABLE A6-2. TAPE CODES

ZONED M11 FORMAT			
Binary		Hex	
Zone	Data	Zone	Data
1001	0000	9	0
1000	0001	8	1
1000	0010	8	2
1000	0011	8	3
1000	0100	8	4
1001	0101	9	5
1001	0110	9	6
1001	0111	9	7
1001	1000	9	8
1001	1001	9	9
1001	1010	9	A
1001	1011	9	B
1001	1100	9	C
1001	1101	9	D
1001	1110	9	E
1001	1111	9	F

APPENDIX 7

Revision Information to OS/32 Library Loader R01

Revision R01 of the OS/32 Library Loader includes a map sort routine command AMAP (LU) to present in alphabetic sequence the memory map of all symbols encountered during a load process. (Program labels, entry points, common blocks, undefined symbols).

This revision has the following operating commands:

FF (forward file mark), BF (backward file mark), FR (forward space record), BR (backspace record), so the user can directly control a device such as Magnetic Tape. This revision eliminates the rewind capability in the FIND, TABLE and EDIT commands available in the OS/32 Library Loader R00.

The following is a list of the Software Change Requests to the R00 version of the OS/32 Library Loader that are in the R01 version of the OS/32 Library Loader.

(1) A Program must be biased on doubleword boundary to run on the 8/32. The Library Loader now biases programs on a doubleword boundary.

(2) When a device is write protected and a write file mark (WF) command is attempted, an I/O error status will be given in the R01 version.

(3) While editing in the batch mode, if an End of File is found, the Loader will print a message EOF and fetch the next command.

R02 has all implicit positioning removed. R01 Bulk Storage procedures no longer apply.

R02 supports the LINKFILE command. This is an internally iterated LINK command.

```
      32-BIT RELOCATING LOADER   03-067M91R02                PAGE   1

PROG= RELLDR     ASSEMBLED BY CAL 03-066R04-01 (16-BIT)

                                    1          SCRAT                                                    L3200030
                                    2          CROSS                                                    L3200040
                                    3          TARGT  32                                                L3200050
                                    4          NORX3                                                    L3200060
000000I                             5          SQCHK                                                    L3200080
                                    6  * INTERDATA 32-BIT RELOCATING LOADER                             L3200100
          0000 0000                 7  R0      EQU    0                                                 L3200110
          0000 0001                 8  R1      EQU    1                                                 L3200120
          0000 0002                 9  R2      EQU    2                                                 L3200130
          0000 0003                10  R3      EQU    3                                                 L3200140
          0000 0004                11  BYTE    EQU    4                                                 L3200150
          0000 0005                12  PICK    EQU    5                                                 L3200160
          0000 0006                13  SEQNUM  EQU    6                                                 L3200170
          0000 0007                14  ONE     EQU    7                                                 L3200180
          0000 0008                15  TWO     EQU    8                                                 L3200190
          0000 0009                16  FOUR    EQU    9                                                 L3200200
          0000 000A                17  A       EQU    10                                                L3200210
          0000 000B                18  B       EQU    11                                                L3200220
          0000 000C                19  C       EQU    12                                                L3200230
          0000 000D                20  D       EQU    13                                                L3200240
          0000 000E                21  E       EQU    14                                                L3200250
          0000 000F                22  ABSF    EQU    15                                                L3200260
          0000 0078                23  BINDV   EQU    X'78'                                             L3200270
          0000 0002                24  RTN     EQU    R2                                                L3200280
          0000 000D                25  DEV     EQU    D                                                 L3200290
          0000 000A                26  AC1     EQU    A                                                 L3200300
          0000 0001                27  DAT     EQU    R1                                                L3200310
          0000 000E                28  CBA     EQU    E                                                 L3200320
          0000 000B                29  CRB     EQU    B                                                 L3200330
          0000 000C                30  ZERO    EQU    C                                                 L3200340
          0000 1000                31  RELPT   EQU    X'1000'                                           L3200350
000000I   2304                     32  BOOT,0  BS     BOOT                                              L3200360
000002I   0000                     33         DB     0,0          POWER FAIL POINTERS                  L3200370
000004I   0000 0000                34         DC     0            POWER FAIL POINTERS                  L3200380
          0000 0008I               35  BOOT    EQU    *                                                 L3200390
000008I   2420                     36         LIS    R2,0                                              L3200400
00000AI   9512                     37         EPSR   R1,R2                                             L3200410
00000CI   D310 0078                38         LB     R1,BINDV     LOAD BINARY DEVICE ADDR              L3200420
000010I   C310 0004                39         THI    R1,X'0004'   IS IT A PTR OR TTYR?          R02 L3200421
000014I   4330 8020                40         BZ     RDBLK        YES,SKIP BACKSPACE           R02 L3200422
000018I   9D12                     41  MTNCHK  SSR    R1,R2        SENSE STATUS                 R02 L3200423
00001AI   C320 0010                42         THI    R2,X'0010'   NO MOTION AND EOM?           R02 L3200424
00001EI   2233                     43         BZS    MTNCHK       NO,WAIT                      R02 L3200425
000020I   DE10 801C                44         OC     R1,BKSPC     BACKSPACE FOR MT OR CASSETTE R02 L3200426
000024I   9D12                     45  MTNCHK1 SSR    R1,R2        SENSE STATUS                 R02 L3200427
000026I   C320 0010                46         THI    R2,X'0010'   NO MOTION AND EOM?           R02 L3200428
00002AI   2233                     47         BZS    MTNCHK1      NO,WAIT                      R02 L3200429
00002CI   C820 0F80                48         LHI    R2,RELPT-X'50' OFFSET RELPT FOR BACKSPACE   R02 L3200430
000030I   5020 8014                49         ST     R2,PARAM     STORE IN PARAM BLOACK        R02 L3200433
000034I   DE10 0079                50         OC     R1,BINDV+1   READ                         R02 L3200435
000038I   D710 800C                51  RDBLK   RB     R1,PARAM     READ DATA                    R02 L3200436
00003CI   4300 1000                52         B      RELPT        CONTINUE                     R02 L3200440
000040I   9100                     53  BKSPC   DB     X'91',0      *                            R02 L3200450
000042I                            54         DO     BOOT,0-*+X'48' *                                 L3200453
```

```
000042I  00                   55            DB      0                     *                                      L3200455
000043I  00                   55            DB      0                     *                                      L3200455
000044I  00                   55            DB      0                     *                                      L3200455
000045I  00                   55            DB      0                     *                                      L3200455
000046I  00                   55            DB      0                     *                                      L3200455
000047I  00                   55            DB      0                     *                                      L3200455
000048I  0000 1000            56   PARAM    DC      RELPT,BUFF-TFIND+RELPT                                       L3200460
00004CI  0000 13EE
000050I  F8A0 0000 8000       57   RELOC    LI      A,Y'8000'             SEARCH TOP OF MEMORY,START=32K   R02   L3200470
000056I  58BA 0000            58   RELOC2   L       B,0(A)                NON-DESTRUCTIVE                  R02   L3200480
00005AI  50AA 0000            59            ST      A,0(A)                *                                R02   L3200490
00005EI  59AA 0000            60            C       A,0(A)                *                                R02   L3200500
000062I  4230 8010            61            BNE     TFIND                 *                                R02   L3200510
000066I  50BA 0000            62            ST      B,0(A)                *                                R02   L3200520
00006AI  CAA0 2000            63            AHI     A,X'2000'             BUMP BY 8K                       R02   L3200530
00006EI  F5A0 0004 0000       64            CLI     A,Y'40000'            256KB?                           R02   L3200540
000074I  203F                 65            BNES    RELOC2                NO,LOOP                          R02   L3200550
000076I  08DA                 66   TFIND    LR      D,A                                                          L3200580
000078I  CBA0 0500            67            SHI     A,X'500'              MOVE LOADER TO TOP OF MEM -X'500'      L3200590
00007CI  E6B0 801A            68            LA      B,START                                                      L3200600
000080I  081A                 69            LR      R1,A                                                         L3200610
000082I  48CB 0000            70   MOVE     LH      C,0(B)                MOVE LOOP                              L3200620
000086I  40CA 0000            71            STH     C,0(A)                                                       L3200630
00008AI  26B2                 72            AIS     B,2                                                          L3200640
00008CI  26A2                 73            AIS     A,2                                                          L3200650
00008EI  05DA                 74            CLR     D,A                                                          L3200660
000090I  2037                 75            BNES    MOVE                                                         L3200670
000092I  F800 0000 8000       76            LI      R0,Y'8000'            GO START LOADER AND WAIT               L3200680
000098I  1800                 77            LPSWR   R0                                                           L3200690
         0000 009AI           78   START    EQU     *                                                            L3200700
00009AI  C8A0 0A00            79            LHI     A,X'A00'              INITIALIZE LOC BIAS                    L3200710
00009EI  2304                 80            BS      REDEF1                *                                R02   L3200720
0000A0I  F8A0 0000 0A00       81   REDEF    LI      A,X'A00'              BIAS DEFINITION                        L3200730
0000A6I  26A7                 82   REDEF1   AIS     A,7                   ALIGN ON DOUBLE WORD BOUNDARY    R02   L3200740
0000A8I  C4A0 FFF8            83            NHI     A,X'FFF8'                                                    L3200750
0000ACI  50A0 83A6            84            ST      A,LOC                                                        L3200760
0000B0I  50A0 83A8            85            ST      A,BIAS                                                       L3200770
0000B4I  50A0 83A8            86            ST      A,PTOP                RESET PTOP TO BIAS                     L3200780
0000B8I  07AA                 87            XR      A,A                                                          L3200790
0000BAI  50A0 8396            88   CLEAR    ST      A,LOCX                                                       L3200800
0000BEI  0766                 89   CONT     XR      SEQNUM,SEQNUM         CLEAR SEQNUM                           L3200810
0000C0I  24F6                 90            LIS     ABSF,6                SET REL MODE                           L3200820
0000C2I  2471                 91            LIS     ONE,1                 INITIALIZE CONSTANT                    L3200830
                              92   * SKIP  NO. FILEMARKS ENTERED ON SWITCHES                                     L3200840
0000C4I  D3D0 0078            93            LB      DEV,BINDV             GET DEVICE ADDRESS                     L3200850
0000C8I  C3D0 0004            94            THI     DEV,4                 MAG DEVICE ?                           L3200860
0000CCI  233E                 95            BZS     NEXT                  NO                                     L3200870
0000CEI  997B                 96            RHR     ONE,CRB               GET NUMBER OF FILE MARKS               L3200880
0000D0I  94BB                 97            EXBR    CRB,CRB                                                      L3200890
0000D2I  C4B0 00FF            98   FF       NHI     CRB,X'FF'             REMOVE LSB                             L3200900
0000D6I  2339                 99            BZS     NEXT                  IF ZERO, NO SKIP                       L3200910
0000D8I  4130 834C           100   SKPFM    BAL     R3,MTCHK              WAIT NO MOTION                         L3200920
0000DCI  DED0 836A           101            OC      DEV,SKIP              SKIP A FILE MARK                       L3200930
0000E0I  27B1                102            SIS     CRB,1                 DECR FILE MARK COUNT                   L3200940
0000E2I  2035                103            BNZS    SKPFM                 CONTINUE                               L3200950
0000E4I  4130 8340           104            BAL     R3,MTCHK                                                     L3200960
```

```
          0000 00E8I          105  JUMP   EQU  *                                                    L3200970
0000E8I   2761                106  NEXT   SIS  SEQNUM,1          DEC SEQ COUNT                   L3200980
0000EAI   2485                107         LIS  TWO,5                                               L3200990
0000ECI   4120 8244           108         BAL  R2,INPUT           INPUT A RECORD                  L3201000
0000F0I   07CC                109         XR   C,C               CLEAR FOR CHECKSUM              L3201010
0000F2I   C8A0 007C           110         LHI  A,124              COMPUTE CHECKSUM                L3201020
0000F6I   47CA 836A           111  CKIT   XH   C,BUFF(A)         XH EM ALL                       L3201030
0000FAI   27A2                112         SIS  A,2                                                 L3201040
0000FCI   2213                113         BNMS CKIT               AND WHEN DONE,RESULT SHOULD     L3201050
0000FEI   26C1                114         AIS  C,1                BE ZERO                         L3201060
000100I   4230 808A           115         BNZ  CERR                                                L3201070
000104I   4560 835C           116         CLH  SEQNUM,BUFF        COMPARE TO SEQNUM               L3201080
000108I   4230 8086           117         BNE  SERR                                                L3201090
00010CI   0755                118         XR   PICK,PICK          START ITEM FETCH                L3201100
00010EI   C550 007A           119  LOOP   CLHI PICK,122           DONE?                           L3201110
000112I   4380 FFD2           120         BNL  NEXT               YES                             L3201120
000116I   D3A5 834E           121         LB   A,BUFF+4(PICK)     GET ITEM                        L3201130
00011AI   0A57                122         AR   PICK,ONE           BUMP INDEX                      L3201140
00011CI   08EA                123         LR   E,A               SAVE                            L3201150
00011EI   C5E0 0020           124         CLHI E,X'20'            ABS ITEM ?                      L3201160
000122I   4380 81F0           125         BNL  ARSDAT             YES                             L3201170
000126I   C5E0 001B           126         CLHI E,X'1B'                                             L3201180
00012AI   4380 8050           127         BNL  LERR                                                L3201190
00012EI   0AAA                128         AR   A,A               VECTOR                          L3201200
000130I   48BA 8004           129         LH   B,JTAB(A)                                           L3201210
000134I   430B FFB0           130         B    JUMP(B)            AND GO DO IT                    L3201220
000138I   0000                131  JTAB   DC   Z(NEXT-JUMP),Z(END-JUMP),Z(RESET-JUMP),Z(LERR-JUMP)  L3201230
00013AI   00CA
00013CI   0086
00013EI   0096
000140I   0104                132         DC   Z(PROADR-JUMP),Z(LERR-JUMP),Z(PROADR-JUMP),Z(LERR-JUMP)  L3201240
000142I   0096
000144I   0104
000146I   0096
000148I   012E                133         DC   Z(REL2B-JUMP),Z(LERR-JUMP),Z(REL4B-JUMP),Z(LERR-JUMP)  L3201250
00014AI   0096
00014CI   014A
00014EI   0096
000150I   0094                134         DC   Z(EXTRN-JUMP),Z(ENTRY-JUMP),Z(LERR-JUMP),Z(LABEL-JUMP)  L3201260
000152I   008E
000154I   0096
000156I   008C
000158I   0096                135         DC   Z(LERR-JUMP),Z(AR33-JUMP),Z(LDX-JUMP),Z(RFIN-JUMP)  L3201270
00015AI   0172
00015CI   00F2
00015EI   0190
000160I   01A2                136         DC   Z(DFIN-JUMP),Z(LERR-JUMP),Z(AR33-JUMP)              L3201280
000162I   0096
000164I   0172
000166I   0096                137         DC   Z(LERR-JUMP),Z(SEGLEN-JUMP),Z(FCHAIN-JUMP)          L3201290
000168I   008A
00016AI   01B2
00016CI   01E4                138         DC   Z(HCHAIN-JUMP)                                      L3201300
00016EI   0766                139  RESET  XR   SEQNUM,SEQNUM      RESET REQUENCE NUMBER           L3201310
000170I   2304                140         BS   ENTRY+2                                             L3201320
000172I   2752                141  SEGLEN SIS  PICK,2             SKIP SEGLEN ITEM                L3201330
```

```
0001741  2754              142  LABEL   SIS   PICK,4                                    L3201340
0001761  265C              143  ENTRY   AIS   PICK,12          SKIP OVER ENTRY ITEM AND PROCEED  L3201350
0001781  4300 FF92         144          B     LOOP                                      L3201360
00017C1  265C              145  EXTRN   AIS   PICK,12          SKIP OVER EXTRN ITEM AND HALT     L3201370
00017E1  E6A0 FF8C         146  LERR    LA    A,LOOP                                    L3201380
0001821  DE70 822C         147          OC    ONE,ASMB                                  L3201390
0001861  9A7E              148          WDR   ONE,E                                     L3201400
0001881  LA70 FF49         149          WD    ONE,FF+3                                  L3201410
00018C1  230F              150          BS    PSW                                       L3201420
00018E1  24A1              151  CERR    LIS   A,1             CHECKSUM ERROR            L3201430
0001901  2306              152          BS    ERROR                                     L3201440
0001921  24A2              153  SERR    LIS   A,2             SEQUENCE ERROR            L3201450
0001941  2304              154          BS    ERROR                                     L3201460
0001961  24A3              155  ADER    LIS   A,3             ADDRESS ERROR             L3201470
0001981  2302              156          BS    ERROR                                     L3201480
00019A1  24A4              157  RELP    LIS   A,4             REF DEF LOOP ERROR        L3201490
00019C1  DE70 8212         158  ERROR   OC    ONE,ASMB                                  L3201500
0001A01  9A7A              159          WDR   ONE,A                                     L3201510
0001A21  DA70 8007         160          WD    ONE,PSW+3                                 L3201520
0001A61  E6A0 FF40         161          LA    A,NEXT+2                                  L3201530
0001AA1  F896 0000 8000    162  PSW     LI    FOUR,X'8000'                              L3201540
0001B01  1809              163          LPSWR FOUR                                      L3201550
0001B21  5800 82AA         164  END     L     R0,PTOP          PROGRAM END              L3201560
0001B61  2607              165          AIS   R0,7             ALIGN ON DOUBLE WORD BOUNDARY  L3201570
0001B81  C400 FFF8         166          NHI   R0,X'FFF8'                                L3201580
0001BC1  5000 829C         167          ST    R0,BIAS          SET NEW BIAS AND LOC      L3201590
0001C01  5000 8294         168          ST    R0,LOC                                    L3201600
0001C41  07AA              169          XR    A,A                                       L3201610
0001C61  DE70 81E8         170          OC    ONE,ASMB                                  L3201620
0001CA1  987A              171          WHR   ONE,A            ZERO DISPLAY              L3201630
0001CC1  58A0 8284         172          L     A,LOCX           TRANSFER IF END/TRANSFER SPECIFIED  L3201640
0001D01  023A              173          BNZR  A                                         L3201650
0001D21  E6A0 FEE8         174          LA    A,CONT                                    L3201660
0001D61  4300 FFD0         175          B     PSW              GO WAIT                  L3201670
0001DA1  D3F5 828A         176  LDX     LB    ABSF,BUFF+4(PICK) PROCESS XFER ADDR       L3201680
0001DE1  0A57              177          AR    PICK,ONE                                  L3201690
0001E01  4130 8106         178          BAL   R3,GETT          GET THE ADDRESS           L3201700
0001E41  5000 826C         179          ST    0,LOCX           SAVE IT                   L3201710
0001E81  4300 FF22         180          B     LOOP             CONT                      L3201720
0001EC1  08FE              181  PROADR  LR    ABSF,E           PROCESS PROGRAM ADDRESS   L3201730
0001EE1  4130 80F8         182          BAL   R3,GETT          GET 3 BYTE ADDR           L3201740
0001F21  E630 FEA4         183  LDL0    LA    R3,START         GET START ADDR OF LDR     L3201750
0001F61  05D3              184          CLR   0,R3             LESS ?                    L3201760
0001F81  2186              185          BLS   LDLX                                      L3201770
0001FA1  E630 82E4         186          LA    R3,LTOP          GET TOP OF LDR            L3201780
0001FE1  05D3              187          CLR   0,R3             LESS?                     L3201790
0002001  4280 FF92         188          BL    ADER                                      L3201800
0002041  5000 8250         189  LDLX    ST    0,LOC            STORE NEW LOC             L3201810
0002081  5500 8254         190  LDL1    CL    0,PTOP           NOT LESS THAN PTOP        L3201820
00020C1  2183              191          BLS   LOOP4                                     L3201830
00020E1  5000 824E         192          ST    0,PTOP           STORE NEW PTOP            L3201840
0002121  4300 FEF8         193  LOOP4   B     LOOP             CONT                      L3201850
0002161  4120 80EC         194  REL2B   BAL   R2,WORDH         GET * BYTES               L3201860
00021A1  5AD0 823E         195          A     0,BIAS           ADD BIAS                  L3201870
00021E1  4280 FF5C         196          BC    LERR                                      L3201880
0002221  58C0 8232         197          L     C,LOC            GET LOC                   L3201890
```

```
000226I 40DC 0000      198          STH   D,0(C)          STORE ADDRESS                    L3201900
00022AI 26C2           199          AIS   C,2             BUMP LOC                         L3201910
00022CI 08DC           200 REL2B2   LR    D,C                                            L3201920
00022EI 4300 FFC0      201          B     LOL0            GO PROCESS NEW LOC               L3201930
000232I 4120 80D0      202 REL4B    BAL   R2,WORDH        GET 2 BYTES                      L3201940
000236I 08BD           203          LR    B,D                                            L3201950
000238I EDB0 0010      204          SLL   B,16            SHIFT OVER 16                    L3201960
00023CI 4120 80C6      205          BAL   R2,WORDH        GET 2 MORE BYTES                 L3201970
000240I 06DB           206          OR    D,B             COMBINE                          L3201980
000242I 5AD0 8216      207          A     D,BIAS          ADD BIAS                         L3201990
000246I 58C0 820E      208          L     C,LOC           GET OC AND STORE                 L3202000
00024AI 40DC 0002      209          STH   D,2(C)                                           L3202010
00024EI 34DD           210          EXHR  D,D                                              L3202020
000250I 40DC 0000      211          STH   D,0(C)                                           L3202030
000254I 26C4           212          AIS   C,4             BUMP LOC AND PROCESS NEW LOC     L3202040
000256I 4300 FFD2      213          B     REL2B2                                           L3202050
00025AI 4120 80A8      214 AR33     BAL   R2,WORDH        GET ADS BYTES                    L3202060
00025EI 58C0 81F6      215          L     C,LOC           STORE AND BUMP LOC AND           L3202070
000262I 40DC 0000      216          STH   D,0(C)          PROCESS REST AS 4 BYTE REL       L3202080
000266I 26C2           217          AIS   C,2                                              L3202090
000268I 50C0 81EC      218          ST    C,LOC                                            L3202100
00026CI C5E0 0016      219          CLHI  E,X'16'                                          L3202110
000270I 4330 FFA2      220          BE    REL2B                                            L3202120
000274I 4300 FFBA      221          B     REL4B                                            L3202130
000278I D3F5 81EC      222 RFIN     LB    ABSF,BUFF+4(PICK) PROCESS REF ADR CHAIN BEGINNING L3202140
00027CI 2651           223          AIS   PICK,1           GET MODE                       L3202150
00027EI 4130 8068      224          BAL   R3,GETT         GET ADDR                         L3202160
000282I 50D0 81C6      225          ST    D,REF           SAVE                             L3202170
000286I 4300 FE84      226          B     LOOP            CONT                             L3202180
00028AI D3F5 81DA      227 DFIN     LB    ABSF,BUFF+4(PICK) PROCESS DEF ADDR AND DO CHAINING L3202190
00028EI 2651           228          AIS   PICK,1           GET MODE                       L3202200
000290I 4130 8056      229          BAL   R3,GETT         GETT ADDR                        L3202210
000294I 50D0 81B8      230          ST    D,DEF                                            L3202220
000298I 2209           231          BS    DFIN-4                                           L3202230
00029AI 58D0 81B2      232 FCHAIN   L     D,DEF                                            L3202240
00029EI 082D           233          LR    R2,D                                             L3202250
0002A0I EC20 0010      234          SRL   R2,16                                            L3202260
0002A4I 58E0 81A4      235          L     E,REF           GET REF POINT                    L3202270
0002A8I 73CE 0002      236 CH1      LHL   C,2(E)          GET NEXT CHAIN ENTRY             L3202280
0002ACI D33E 0001      237          LB    R3,1(E)                                          L3202290
0002B0I D22E 0001      238          STB   R2,1(E)         STORE PARTS OF DEF ADDR: 1BYTE   L3202300
0002B4I 40DE 0002      239          STH   D,2(E)          2BYTES                           L3202310
0002B8I 3433           240          EXHR  R3,R3                                            L3202320
0002BAI 06C3           241          OR    C,R3                                             L3202330
0002BCI 08EC           242          LR    E,C             END OF CHAIN?                    L3202340
0002BEI 4330 FE4C      243 CH10     BZ    LOOP                                             L3202350
0002C2I 05CD           244          CLR   C,D             REF/DEF LOOP                     L3202360
0002C4I 4330 FED2      245 CH11     BE    RELP            YES                              L3202370
0002C8I 4300 FFDC      246          B     CH1                                              L3202380
0002CCI 58E0 817C      247 HCHAIN   L     E,REF           DO HALFWORD CHAIN                L3202390
0002D0I 58D0 817C      248          L     D,DEF                                            L3202400
0002D4I 73CE 0000      249 CH2      LHL   C,0(E)                                           L3202410
0002D8I 40DE 0000      250          STH   D,0(E)                                           L3202420
0002DCI 08EC           251          LR    E,C                                              L3202430
0002DEI 4330 FE2C      252          BZ    LOOP                                             L3202440
0002E2I 05CD           253          CLR   C,D                                              L3202450
```

```
0002E4I 4330 FFDC      254          BE    CH11                                          L3202460
0002E8I 220A           255          BS    CH2                                           L3202470
0002EAI D305 817A      256  GETT    LB    R0,BUFF+4(PICK)  GET 3 BYTE ADDR AND ADD IN BIAS L3202480
0002EEI 2651           257          AIS   PICK,1                                       L3202490
0002F0I 4120 8012      258          BAL   R2,WORDH         GET 200 2 BYTES             L3202500
0002F4I ED00 0010      259          SLL   R0,16                                        L3202510
0002F8I 06D0           260          OR    D,R0                                          L3202520
0002FAI C5F0 0004      261          CLHI  ABSF,X'4'        ABS?                        L3202530
0002FEI 0333           262          BER   R3                                          L3202540
000300I 5AD0 8158      263          A     D,BIAS           ADD IN BIAS                 L3202550
000304I 0303           264          BR    R3                                          L3202560
000306I D3D5 815E      265  WORDH   LB    D,BUFF+4(PICK)   GET 2BYTES FROM BUFFER     L3202570
00030AI D3C5 815B      266          LB    C,BUFF+5(PICK)                              L3202580
00030EI 2652           267          AIS   PICK,2                                      L3202590
000310I 11D8           268          SLLS  D,8                                         L3202600
000312I 06DC           269          OR    D,C                                         L3202610
000314I 0302           270          BR    R2                                          L3202620
000316I 58D0 813E      271  ABSDAT  L     D,LOC            PROCESS ADS DATA            L3202630
00031AI CBE0 001F      272          SHI   E,X'1F'          ADJUST VECTOR               L3202640
00031EI 0AEE           273          AR    E,E                                         L3202650
000320I D3C5 8144      274  ABSLP   LB    C,BUFF+4(PICK)   MOVE BYTES                 L3202660
000324I D2CD 0000      275          STB   C,0(D)                                      L3202670
000328I 2651           276          AIS   PICK,1                                      L3202680
00032AI 26D1           277          AIS   D,1                                         L3202690
00032CI 27E1           278          SIS   E,1                                         L3202700
00032EI 2037           279          BNZS  ABSLP                                       L3202710
000330I 4300 FEBE      280          B     LDL0             SET NEW LOC WHEN DONE       L3202720
000334I D3D0 0078      281  INPUT   LB    DEV,BINDV        GET INPUT DEVICE            L3202730
000338I C3D0 0001      282          THI   DEV,1            TTY?                        L3202740
00033CI 2139           283          BNZS  IN1              NO                          L3202750
00033EI DED0 8104      284  INP1    OC    DEV,TWRT         YES,WRITE MODE COMMAND      L3202760
000342I 2042           285          BOS   INP1                                        L3202770
000344I 4130 80D8      286          BAL   R3,DEVCHK        WAIT FOR BUSY LOW           L3202780
000348I DAD0 80FC      287  IN0     WD    DEV,XON          START TAPE READER     R02   L3202790
00034CI 2042           288          BOS   IN0              *                     R02   L3202800
00034EI DED0 0079      289  IN1     OC    DEV,BINDV+1      'SET WRITE MODE             L3202810
000352I 2042           290          BOS   IN1              *                     R02   L3202820
000354I C3D0 0004      291          THI   DEV,4            MAG TAPE DEVICE?            L3202830
000358I 4230 8088      292          BNZ   MTCAS            YES                         L3202840
00035CI 07EE           293  IN2     XR    CBA,CBA          CLEAR BUFFER POINTER        L3202850
00035EI 40E0 80E0      294          STH   CBA,CFLG         AND FLAGS                   L3202860
000362I 40E0 80DE      295          STH   CBA,BFLG                                     L3202870
000366I 08CE           296          LR    ZERO,CBA                                    L3202880
000368I 4130 80B4      297  READIT  BAL   R3,DEVCHK        CHECK FOR BUSY              L3202890
00036CI 9BD1           298  RD,DTA  RDR   DEV,DAT          GET DATA              R02   L3202900
00036EI 2041           299          BOS   RD,DTA           *                     R02   L3202910
000370I 9A71           300          WDR   ONE,DAT          SEND TO LIGHTS              L3202920
000372I 48A0 80CC      301          LH    AC1,CFLG         GET ZONE FLAG               L3202930
000376I 4220 8042      302          BP    STORE            BRANCH IF NON-ZONED         L3202940
00037AI 2118           303          BMS   TESTB            BRANCH IF ZONED             L3202950
00037CI C510 00F0      304          CLHI  DAT,X'F0'        CHECK IF OUT                L3202960
000380I 2135           305          BNES  TESTB            NOT ZONED                   L3202970
000382I 4070 80BC      306          STH   ONE,CFLG         ZONED SET FLAG              L3202980
000386I 4300 FFDE      307          B     READIT           CONTINUE                    L3202990
00038AI 08A1           308  TESTB   LR    AC1,DAT          GET DATA AND CHECK          L3203000
00038CI C4A0 000F      309          NHI   AC1,15           FOR VADID ZONE              L3203010
```

```
000390I D3AA 809E          310          LB    AC1,ZTAB(AC1)                                    L3203020
000394I 051A               311          CLR   DAT,AC1                                          L3203030
000396I 4230 FFCE          312          BNE   READIT          INVALID ZONE, SKIP CHAR          L3203040
00039AI 48A0 80A6          313          LH    AC1,BFLG        CHECK FLIP SWITCH                L3203050
00039EI 4230 8010          314          BNZ   ASMB                                             L3203060
0003A2I 4070 809E          315          STH   ONE,BFLG        SET IT                           L3203070
0003A6I 08B1               316          LR    CRB,DAT                                          L32030A0
0003A8I 11B4               317          SLLS  CRB,4           GET TOP 4 BITS                   L3203090
0003AAI 4060 8094          318          STH   SEQNUM,CFLG     SET ZONE FLAG                    L3203100
0003AEI 4300 FFB6          319          B     READIT                                           L3203110
0003B2I 40C0 808E          320  ASMB    STH   ZERO,BFLG       RESET FLIP SWITCH                L3203120
0003B6I C410 000F          321          NHI   DAT,15                                           L3203130
0003BAI 061B               322          OR    DAT,CRB         OR IN OTHER 4 BITS               L3203140
0003BCI D21E 80A4          323  STORE   STB   DAT,BUFF(CBA)    STORE A BYTE                     L3203150
0003C0I 0AE7               324          AR    CBA,ONE         BUMP POINTER                     L3203160
0003C2I C5E0 007E          325          CLHI  CBA,126         DONE?                            L3203170
0003C6I 4280 FF9E          326          BL    READIT          NO,LOOP                          L3203180
0003CAI C3D0 0001          327          THI   DEV,1           TTY ?                            L3203190
0003CEI 2139               328          BNZS  DVROUT          NO,BRANCH                        L3203200
0003D0I DED0 8072          329  TTYOFF  OC    DEV,TWRT        TAPE OFF                         L3203210
0003D4I 2042               330          BOS   TTYOFF          *                    R02         L3203220
0003D6I 9DD1               331          SSR   DEV,DAT                                          L3203230
0003D8I 20F1               332          BTBS  15,1                                             L3203240
0003DAI DAD0 806B          333  TTYOFF1 WD    DEV,XOFF        *                    R02         L3203250
0003DEI 2042               334          BOS   TTYOFF1         *                    R02         L3203260
0003E0I 2485               335  DVROUT  LIS   TWO,5           SET RETRY COUNTER                L3203270
0003E2I 0302               336          BR    R2              RETURN                           L3203280
0003E4I 9DD1               337  MTCAS   SSR   DEV,DAT                                          L3203290
0003E6I 2325               338          BNPS  MTCAS1          EOM? NO                          L3203300
0003E8I 4130 803C          339          BAL   R3,MTCHK        CHECK NO MOTION                  L3203310
0003ECI 4300 FF5E          340          B     IN1                                              L3203320
0003F0I E640 8070          341  MTCAS1  LA    BYTE,BUFF                                        L3203330
0003F4I E650 80E9          342          LA    BYTE+1,BUFF+125                                  L3203340
0003F8I 97D4               343          RBR   DEV,BYTE        READ A RECORD                    L3203350
0003FAI 4130 8022          344          BAL   R3,DEVCHK                                        L3203360
0003FEI 4320 801E          345          BNP   DEVCHK                                           L3203370
000402I C310 00A0          346          THI   DAT,X'A0'                                        L3203380
000406I 4330 FFD6          347          BZ    DVROUT                                           L3203390
00040AI 4130 801A          348          BAL   R3,MTCHK                                         L3203400
00040EI DED0 8036          349          OC    DEV,BAKREC      BACKSPACE                        L3203410
000412I 4130 8012          350          BAL   R3,MTCHK        NO MOTION WAIT                   L3203420
000416I 0B87               351          SR    TWO,ONE         2 RETRYS?                        L3203430
000418I 4310 FF32          352          BNM   IN1                                              L3203440
00041CI 4300 FD6E          353  MTERR   B     CERR            YES,ERROR                        L3203450
000420I 9DD1               354  DEVCHK  SSR   DEV,DAT         DEVICE STATUS CHECK              L3203460
000422I 2013               355          BMS   MTERR                                           L3203470
000424I 2082               356          BCS   DEVCHK                                          L3203480
000426I 0303               357          BR    R3                                               L3203490
000428I 9DD1               358  MTCHK   SSR   DEV,DAT         MAG TAPE/CASSETTE NO MOTION CHECK L3203500
00042AI C410 0010          359          NHI   DAT,X'10'                                        L3203510
00042EI 2233               360          BZS   MTCHK                                          L3203520
000430I 0303               361          BR    R3                                               L3203530
                           362  * TABLE OF NON-PRINTING ASCII SET                              L3203540
000432I 9081               363  ZTAB    DC    X'9081',X'8283',X'8495',X'9697',X'9899',X'9A9B',X'9C9D' L3203550
000434I 8283
000436I 8495
```

```
000438I  9697
00043AI  9899
00043CI  9A9B
00043EI  9C9D
000440I  9E9F                    364           DC      X'9E9F'                  L3203560
000442I  0000                    365  CFLG     DC      X'0'                     L3203570
000444I  0000                    366  BFLG     DC      X'0'                     L3203580
000446I  98A9                    367  TWRT     DC      X'98A9'                  L3203590
000448I  9193                    368  XON      DC      X'9193'                  L3203600
         0000 0449I              369  XOFF     EQU     XON+1                    L3203610
         0000 0448I              370  BAKREC   EQU     XON                      L3203620
00044AI  A320                    371  SKIP     DC      X'A320'                  L3203630
         0000 044BI              372  MTCLR    EQU     SKIP+1                   L3203640
00044CI                          373           ALIGN   4                        L3203650
00044CI                          374  REF      DS      4                        L3203660
000450I                          375  DEF      DS      4                        L3203670
000454I                          376  LOCX     DS      4                        L3203680
000458I                          377  LOC      DS      4                        L3203690
00045CI                          378  BIAS     DS      4                        L3203700
000460I                          379  PTOP     DS      4                        L3203710
000464I                          380  BUFF     DS      126                      L3203720
0004E2I                          381  LTOP     END                              L3203730
```

NO ERRORS 0 SQUEZ PASSES

CAL 04-01

A	0000 000A	26 57 58 59 59 60 60 62 63 64 66 67 69 71 73 74 79 81 82 83 84 85 86 87 87 88 110 111 112 121 123 128 128 129 146 151 153 155 157 159 161 169 169 171 172 173 174
ABSDAT	0000 0316I	125
ABSF	0000 000F	90 176 181 222 227 261
ABSLP	0000 0320I	279
ABSTOP	0000 0000	
AC1	0000 000A	301 308 309 310 310 311 313
ADC	0000 0004	
ADER	0000 0196I	188
AR33	0000 025AI	135 136
ASMB	0000 03B2I	147 158 170 314
B	0000 000B	29 58 62 68 70 72 129 130 203 204 206
BAKREC	0000 0448I	349
BFLG	0000 0444I	295 313 315 320
BIAS	0000 045CI	85 167 195 207 263
BINDV	0000 0078	38 50 93 281 289
BKSPC	0000 0040I	44
BOOT	0000 0008I	32
BOOT.0	0000 0000I	54
BUFF	0000 0464I	56 111 116 121 176 222 227 256 265 266 274 323 341 342
BYTE	0000 0004	341 342 343
C	0000 000C	30 70 71 109 109 111 114 197 198 199 200 208 209 211 212 215 216 217 218 236 241 242 244 249 251 253 266 269 274 275
CBA	0000 000E	293 293 294 295 296 323 324 325
CERR	0000 018EI	115 353
CFLG	0000 0442I	294 301 306 318
CH1	0000 02A8I	246
CH10	0000 02BEI	
CH11	0000 02C4I	254
CH2	0000 02D4I	255
CKIT	0000 00F6I	113
CLEAR	0000 00BAI	
CONT	0000 00BEI	174
CRB	0000 000B	96 97 97 98 102 316 317 322
D	0000 000D	25 66 74 179 184 187 189 190 192 195 198 200 203 206 207 209 210 210 211 216 225 230 232 233 239 244 248 250 253 260 263 265 268 269 271 275 277
DAT	0000 0001	298 300 304 308 311 316 321 322 323 331 337 346 354 358 359
DEF	0000 0450I	230 232 248
DEV	0000 000D	93 94 101 281 282 284 287 289 291 298 327 329 331 333 337 343 349 354 358
DEVCHK	0000 0420I	286 297 344 345 356
DFIN	0000 028AI	136 231
DVROUT	0000 03E0I	328 347
E	0000 000E	28 123 124 126 148 161 219 235 236 237 238 239 242 247 249 250 251 272 273 273 278
END	0000 01B2I	131

ENTRY 0000 0176I 134 140
ERROR 0000 019CI 152 154 156
EXTRN 0000 017CI 134
FCHAIN 0000 029AI 137
FF 0000 00D2I 149
FOUR 0000 0009 162 163
GETT 0000 02EAI 178 182 224 229
HCHAIN 0000 02CCI 138
IMPTOP 0000 04E2I
IN0 0000 0348I 288
IN1 0000 034EI 283 290 340 352
IN2 0000 035CI
INP1 0000 033EI 285
INPUT 0000 0334I 108
JTAB 0000 0138I 129
JUMP 0000 00E8I 130 131 131 131 131 132 132 132 132 133 133 133 133
134 134 134 134 135 135 135 135 136 136 136 137 137
137 138
LABEL 0000 0174I 134
LADC 0000 0002
LDL0 0000 01F2I 201 280
LDL1 0000 0206I
LDLX 0000 0204I 185
LDX 0000 01DAI 135
LERR 0000 017EI 127 131 132 132 133 133 134 135 136 137 196
LOC 0000 0458I 84 168 189 197 208 215 212 271
LOCX 0000 0454I 88 172 179
LOOP 0000 010EI 144 146 180 193 226 243 252
LOOP4 0000 0212I 191
LTOP 0000 04E2I 186
MOVE 0000 0082I 75
MTCAS 0000 03E4I 292
MTCAS1 0000 03F0I 338
MTCHK 0000 0428I 100 104 339 348 350 360
MTCLR 0000 044BI
MTERR 0000 041CI 355
MTNCHK 0000 0018I 43
MTNCHK1 0000 0024I 47
NEXT 0000 00E8I 95 99 120 131 161
ONE 0000 0007 91 96 122 147 148 149 158 159 160 170 171 177 300
306 315 324 351
PARAM 0000 0048I 49 51
PICK 0000 0005 118 118 119 121 122 141 142 143 145 176 177 222 223
227 228 256 257 265 266 267 274 276
PROADR 0000 01ECI 132 132
PSW 0000 01AAI 150 160 175
PTOP 0000 0460I 86 164 190 192
PURETOP 0000 0000P
R0 0000 0000 76 77 164 165 166 167 168 256 259 260
R1 0000 0001 27 37 38 39 41 44 45 50 51 69
R2 0000 0002 24 36 37 41 42 45 46 48 49 108 194 202 205
214 233 234 238 258 270 336
R3 0000 0003 100 104 178 182 183 184 186 187 224 229 237 240 240
241 262 264 286 297 339 344 348 350 357 361
RD.DTA 0000 036CI 299
RDBLK 0000 0038I 40

READIT	0000	0368I	307	312	319	326			
REDEF	0000	00A0I							
REDEF1	0000	00A6I	80						
REF	0000	044CI	225	235	247				
REL2B	0000	0216I	133	220					
REL2B2	0000	022CI	213						
REL4B	0000	0232I	133	221					
RELOC	0000	0050I							
RELOC2	0000	0056I	65						
RELP	0000	019AI	245						
RELPT	0000	1000	48	52	56	56			
RESET	0000	016EI	131						
RFIN	0000	0278I	135						
RTN	0000	0002							
SEGLEN	0000	0172I	137						
SEQNUM	0000	0006	89	89	106	116	139	139	318
SERR	0000	0192I	117						
SKIP	0000	044AI	101	372					
SKPFM	0000	00D8I	103						
START	0000	009AI	68	183					
STORE	0000	03BCI	302						
TESTB	0000	038AI	303	305					
TFIND	0000	0076I	56	61					
TTYOFF	0000	03D0I	330						
TTYOFF1	0000	03DAI	334						
TWO	0000	0008	107	335	351				
TWRT	0000	0446I	284	329					
WORDH	0000	0306I	194	202	205	214	258		
XOFF	0000	0449I	333						
XON	0000	0448I	287	369	370				
ZERO	0000	000C	296	320					
ZTAB	0000	0432I	310						

```
PROG= BOOT32        ASSEMBLED BY CAL 03-066R04(32-BIT)

          **03074003
          **BOOT32                                                                      B3200010
                  NORX3                                                                 B3200020
          * COPYRIGHT INTERDATA INC. APRIL 1975                                         B3200040
          *                                                                             B3200050
          * THIS PROGRAM IS A LOADER THAT READS A CORE IMAGE OF OS 32 ST OR MT          B3200060
          * FROM A 2.5,10,40,67, OR 252 MEGABYTE DISC.                                  B3200070
          * DEFINITION TABLE:                                                           B3200080
          *                                                                             B3200090
          *    MEMORY LOCATION          CONTENTS                                        B3200100
          *    ------ --------          --------                                        B3200110
          *        X'78'             BINARY INPUT DEVICE ADR                      -     B3200120
          *        X'79'             BINARY DEVICE CODE                                 B3200130
          *        X'7A'             DISC FILE ADDRESS  (FILE CONTAINING OS) -          B3200140
          *        X'7B'             DEVICE CODE (SEE DEVICE TABLE BELOW)    -          B3200150
          *        X'7C'             DISC CONTROLLER ADDRESS                 -          B3200160
          *        X'7D'             EXTENDED SELCH ADDRESS                  -          B3200170
          *        X'7E',X'7F'       OS EXTENSION NUMBER IN HEX                         B3200180
          *    ---------------------------------------------------------------          B3200190
          *                                                                             B3200200
          *    DISC  DEVICE CODES       DISC TYPE                                       B3200210
          *    ----- ------ ----        ----                                            B3200220
          *        X'30'             2.5 MB DISC (FIXED)                                B3200230
          *        X'31'             2.5 MB DISC (REMOVABLE)                            B3200240
          *        X'32'             10 MB DISC  (FIXED)                                B3200250
          *        X'33'             10 MB DISC  (REMOVABLE)                            B3200260
          *        X'34'             40 MB DISC                                         B3200270
          *        X'35'             67 MB DISC                                         B3200280
          *        X'36'             252 MB DISC                                        B3200282
          *                                                                             B3200284
          * THE PROGRAM WILL PERFORM THE FOLLOWING UPON EXECUTION:                      B3200290
          *      1.READ THE VOLUME DESCRIPTOR FROM CYL0,SECTOR0,HEAD0 OF THE            B3200300
          *        DISC                                                                 B3200310
          *      2.EXTRACT FROM THE VOLUME DESCRIPTOR THE STARTING LOGICAL SECT-        B3200320
          *        OR OF THE DIRECTORY                                                  B3200330
          *      3. SEARCH DIRECTORY FOR FILE WITH NAME "OS32XXXX.YYY",                 B3200340
          *        WHERE "XXXX" CAN BE ANY CHARACTERS, AND "YYY" IS THE                 B3200350
          *        EXTENSION SPECIFIED IN X'7E,7F'                                      B3200360
          *      4. IF THE CORRECT OS IS NOT FOUND IN THE DIRECTORY, AND THE            B3200370
          *        EXTENSION = ZERO, THE OS POINTED TO BY THE VOLUME                    B3200380
          *        DESCRIPTOR IS USED (IF ONE EXISTS).                                  B3200390
          *      5. THE LOADER RELOCATES ITSELF TO THE TOP OF THE OS, AND               B3200400
          *        LOADS AND BRANCHES TO THE OS.                                        B3200410
          *                                                                             B3200420
          *                                                                             B3200430
          * SYSGEN PARAMETERS = NONE                                                    B3200440
          *                                                                             B3200450
          * ENTRY/EXTRN DECLARATIONS = NONE                                             B3200460
          *                                                                             B3200470
000000I           COPY   DIR                                                            B3200480
          *                                                                             B3200490
          *DIR                         DEFINED AS DIRECTORY                             MCB05820
          *                                                                             MCB05830
          DIR     STRUC                                                                 MCB05840
                                                                                        MCB05850
```

```
000000                    56  DIR.FNM  DS    8                FILE NAME                              MCB05860
000008                    57  DIR.EXT  DS    4                EXTENTION                    -3B       MCB05870
        0000 000B         58  DIR.VERS EQU   DIR.EXT+3        VERSION                      -1B       MCB05880
00000C                    59  DIR.FLBA DS    4                FIRST LOGICAL BLOCK ADDR               MCB05890
000010                    60  DIR.LLBA DS    4                LAST LOGICAL BLOCK ADDR                MCB05900
000014                    61  DIR.KEYS DS    0                KEYS                                   MCB05910
000014                    62  DIR.WKEY DS    1                WRITE KEY                    -1B       MCB05920
000015                    63  DIR.RKEY DS    1                READ KEY                     -1B       MCB05930
000016                    64  DIR.LRCL DS    2                LOGICAL RECORD                         MCB05940
000018                    65  DIR.DATE DS    4                DATE FILE ALLOCATED                    MCB05950
00001C                    66  DIR.LUSE DS    4                DATE FILE LAST ASSIGNED                MCB05960
000020                    67  DIR.WCNT DS    2                WRITE COUNT                            MCB05970
000022                    68  DIR.RCNT DS    2                READ COUNT                             MCB05980
000024                    69  DIR.ATRB DS    1                ATTRIBUTES                   -1B       MCB05990
000025                    70  DIR.BKSZ DS    1                BLOCK SIZE                   -1B       MCB06000
000026                    71  DIR.INBS DS    1                INDEX BLOCK SIZE                       MCB06010
000027                    72           DS    1                RESERVED                               MCB06020
000028                    73  DIR.CSEC DS    4                CURRENT SECTOR /# LOGICAL RECORDS      MCB06030
00002C                    74           DS    4                RESERVED                               MCB06040
        0000 0003         75  DIRA.ACB EQU   3                                                        MCB06050
        0000 0010         76  DIRA.ACM EQU   X'10'                                                    MCB06060
        0000 0007         77  DIRA.ALB EQU   7                PREALLOCATE BIT(DISCINIT)              MCB06070
        0000 0001         78  DIRA.ALM EQU   X'01'            PREALLOCATE MASK                       MCB06080
        0000 00F4         79  DIR.RSAS EQU   244                                                      MCB06090
000030                    80           ENDS                                                           MCB06100
                          81  * REGISTERS EQUATE                                                     B3200500
        0000 0000         82  U0       EQU   0                RERMAINDER                             B3200510
        0000 0001         83  U1       EQU   1                QUOTIENT                               B3200520
        0000 0002         84  U2SELCH  EQU   2                SLECH ADDRESS                          B3200530
        0000 0003         85  U3CTRL   EQU   3                CONTROLLER ADDRESS                     B3200540
        0000 0004         86  U4DEV    EQU   4                DISC ADDRESS                           B3200550
        0000 0005         87  U5DEVCD  EQU   5                DEVICE CODE                            B3200560
        0000 0006         88  U6LOOP   EQU   6                DISC READ LOOP ADDRESS                 B3200570
        0000 0007         89  U7STARTA EQU   7                VOLUME DESCRIPTOR START ADDRESS        B3200580
        0000 0008         90  U8ENDA   EQU   8                VOLUME DESCRIPTOR ENDING ADDRESS       B3200590
        0000 0009         91  U9CYLN   EQU   9                DISC CYLINDER NUMBER                   B3200600
        0000 000A         92  UAHEADN  EQU   10               DISC HEAD NUMBER                       B3200610
        0000 000B         93  UBSECN   EQU   11               DISC SECTOR NUMBER                     B3200620
        0000 000C         94  UCLSECN  EQU   12               DISC LOGICAL SECTOR NUMBER             B3200630
        0000 000D         95  UDBIAS   EQU   13               TOP OF OS IMAGE BIAS                   B3200640
        0000 000E         96  UELINK   EQU   14               LINK ADDRESS FOR SUBROUTINE CALLS      B3200650
        0000 000F         97  UF       EQU   15               USER REGISTER 15                       B3200660
                          98  *                                                                      B3200670
        0000 0001         99  R1       EQU   1                REG 1                                  B3200680
        0000 0002        100  R2       EQU   2                REG 2                                  B3200690
        0000 0003        101  R3       EQU   3                REG 3                                  B3200700
        0000 0004        102  R4       EQU   4                REG 4                                  B3200710
        0000 007A        103  DEVA     EQU   X'7A'            DISC FILE ADDRESS LOCATION             B3200720
        0000 007B        104  DEVCD    EQU   X'7B'            DEVICE CODE LOCATION                   B3200730
        0000 007C        105  CTRLA    EQU   X'7C'            CONTROLLER ADDRESS LOCATION            B3200740
        0000 007D        106  SELCHA   EQU   X'7D'            SELCH ADDRESS LOCATION                 B3200750
        0000 0060        107  OSINITA  EQU   X'60'            OS INITIALIZATION ADDRESS              B3200760
        0000 1000        108  BOOTADR  EQU   X'1000'          BOOT START ADDRESS                     B3200770
```

```
INTERDATA 32-BIT BOOTSTRAP LOADER   **03-074M91R03A13**   PAGE   3   14:02:40   04/22/76

BOOTSTRAP LOADER / ABSOLUTE SEGMENT 1

000000I                 110            ORG    X'80'                                                  B3200790
000080   2410           111   BOOTLDR  LIS    R1,0                 INITIALIZE II NEW PSW             B3200800
000082   4830 009E      112            LH     R3,INITADR                                             B3200810
000086   4840 00A0      113            LH     R4,INITADR+2                                           B3200820
00008A   4010 0030      114            STH    R1,X'30'             STORENEW PSW IN LOC X'30'         B3200830
00008E   4010 0032      115            STH    R1,X'32'                                               B3200840
000092   4030 0034      116            STH    R3,X'34'                                               B3200850
000096   4040 0036      117            STH    R4,X'36'                                               B3200860
00009A   C200 0030      118            LPSW   X'30'                                                  B3200870
00009E   0000 00A2      119   INITADR  DC     A(INIT)                                                B3200880



         0000 00A2      121   INIT     EQU    *                                                      B3200900
0000A2   D340 0078      122            LB     U4DEV,X'78'          GET BINARY INPUT UNIT NUMBER      B3200910
0000A6   D740 00C8      123   BOOTSTRP RB     U4DEV,BOTRANGE       READ IN SUCCEEDING SEGMENTS       B3200920
0000AA   4300 100E      124            B      BOOTADR+BOOTSTRT-CNV                                   B3200930
0000AE                  125            DO     X'C8'-*                                                B3200940
0000AE   00             126            DB     0                                                      B3200950
0000AF   00             126            DB     0                                                      B3200950
0000B0   00             126            DB     0                                                      B3200950
0000B1   00             126            DB     0                                                      B3200950
0000B2   00             126            DB     0                                                      B3200950
0000B3   00             126            DB     0                                                      B3200950
0000B4   00             126            DB     0                                                      B3200950
0000B5   00             126            DB     0                                                      B3200950
0000B6   00             126            DB     0                                                      B3200950
0000B7   00             126            DB     0                                                      B3200950
0000B8   00             126            DB     0                                                      B3200950
0000B9   00             126            DB     0                                                      B3200950
0000BA   00             126            DB     0                                                      B3200950
0000BB   00             126            DB     0                                                      B3200950
0000BC   00             126            DB     0                                                      B3200950
0000BD   00             126            DB     0                                                      B3200950
0000BE   00             126            DB     0                                                      B3200950
0000BF   00             126            DB     0                                                      B3200950
0000C0   00             126            DB     0                                                      B3200950
0000C1   00             126            DB     0                                                      B3200950
0000C2   00             126            DB     0                                                      B3200950
0000C3   00             126            DB     0                                                      B3200950
0000C4   00             126            DB     0                                                      B3200950
0000C5   00             126            DB     0                                                      B3200950
0000C6   00             126            DB     0                                                      B3200950
0000C7   00             126            DB     0                                                      B3200950
         0000 1270      127   S1END    EQU    BOOTADR+LENGTH                                         B3200960
0000C8   0000 1000      128   BOTRANGE DC     BOOTADR                                                B3200970
0000CC   0000 1270      129            DC     S1END                                                  B3200980
```

```
INTERDATA 32-BIT BOOTSTRAP LOADER  **03-074M91R03A13**   PAGE   4   14:02:41  04/22/76

BOOTSTRAP LOADER / RELOCATABLE SEGMENT 2

                                 131  *--------------------------------MAIN LOGIC STARTS HERE          B3201000
          0000 00D0              132  SEGMENT2 EQU    *                                              B3201010
                                 133  *                                                              B3201020
                                 134  *........CONVERT HEX TO ASCII SUBROUTINE                       B3201030
                                 135  *                                                              B3201040
 0000D0   CAF0 0030              136  CNV       AHI   UF,X'30'                                       B3201050
 0000D4   C5F0 003A              137             CLHI  UF,X'3A'                                       B3201060
 0000D8   028E                   138             BLR   UELINK                                         B3201070
 0000DA   26F7                   139             AIS   UF,7                                           B3201080
 0000DC   030E                   140             BR    UELINK                                         B3201090
                                 141  *                                                              B3201100
 0000DE   E6D0 0F30              142  BOOTSTRT  LDAI  UDBIAS,BOOTADR-SEGMENT2                        B3201110
 0000E2   D1E0 0078              143             LM    UELINK,X'78'                                   B3201120
 0000E6   D0ED 0340              144             STM   UELINK,SAVE(UDBIAS)                          B3201130
 0000EA   D3F0 007E              145             LB    UF,X'7E'          GET 1ST BYTE OF EXTENSION    B3201140
 0000EE   C4F0 000F              146             NHI   UF,X'F'           REMOVE TOP NIBBLE            B3201150
 0000F2   41ED 00D0              147             BAL   UELINK,CNV(UDBIAS) CONVERT TO ASCII            B3201160
 0000F6   D2FD 0325              148             STB   UF,VERS+1(UDBIAS)                            B3201170
 0000FA   D3F0 007F              149             LB    UF,X'7F'          GET 2ND CHAR OF EXT.         B3201180
 0000FE   10F4                   150             SRLS  UF,4                                         B3201190
 000100   41ED 00D0              151             BAL   UELINK,CNV(UDBIAS) CONVERT 2ND TP ASCII        B3201200
 000104   D2FD 0326              152             STB   UF,VERS+2(UDBIAS)                            B3201210
 000108   D3F0 007F              153             LB    UF,X'7F'                                     B3201220
 00010C   C4F0 000F              154             NHI   UF,X'F'                                      B3201230
 000110   41ED 00D0              155             BAL   UELINK,CNV(UDBIAS) CONVERT 3RD CHAR            B3201240
 000114   D2FD 0327              156             STB   UF,VERS+3(UDBIAS)                            B3201250
 000118   D320 007D              157             LB    U2SELCH,SELCHA    GET SELCH ADR.               B3201260
 00011C   D330 007C              158             LB    U3CTRL,CTRLA      GET CTRL ADR                 B3201270
 000120   D340 007A              159             LB    U4DEV,DEVA        GET DEV ADR                  B3201280
 000124   2450                   160             LIS   U5DEVCD,0         TABLE INDEX 10 MB OR SMALLER B3201284
 000126   D3F0 007B              161             LB    UF,DEVCD                                     B3201286
 00012A   CBF0 0033              162             SHI   UF,X'33'          10 MB OR SMALLER ?           B3201288
*00012E   2323                   163             BNP   BOSTRT1           B IF YES                     B3201290
 000130   11F1                   164             SLLS  UF,1                                         B3201292
 000132   085F                   165             LR    U5DEVCD,UF        USE DCOD FOR INDEX           B3201294
          0000 0134              166  BOSTRT1   EQU   *                                              B3201296
 000134   E67D 0350              167             LDAI  U7STARTA,VDBUF(UDBIAS)   GET VD START ADR     B3201300
 000138   E68D 044F              168             LDAI  U8ENDA,VDBUFE(UDBIAS)   GET VD END ADR       B3201310
 00013C   07CC                   169             XAR   UCLSECN,UCLSECN   SET LOGICAL SECTOR NO TO 0   B3201320
 00013E   41ED 01FC              170             BAL   UELINK,READ(UDBIAS) READ VD                  B3201330
 000142   D1ED 035C              171             LM    UELINK,VD.OSP(UDBIAS) SAVE OS INFO FROM V.D  B3201340
 000146   D0ED 0348              172             STM   UELINK,OSP(UDBIAS)                           B3201350
                                 173  *                                                              B3201360
                                 174  *........SEARCH DIRECTORY FOR PROPER OS REVISION               B3201370
                                 175  *                                                              B3201380
 00014A   58CD 0358              176             LDA   UCLSECN,VD.FDP(UDBIAS)                       B3201390
 00014E   433D 019C              177             BZ    XX.070(UDBIAS)    B IF NO DIRECTORY          B3201400
 000152   E67D 0350              178  XX.000    LDAI  U7STARTA,VDBUF(UDBIAS)                       B3201410
 000156   41ED 01FC              179             BAL   UELINK,READ(UDBIAS)                          B3201420
 00015A   E6ED 0354              180             LDAI  UELINK,VDBUF+4(UDBIAS)   POINT TO FIRST ENTRY B3201430
 00015E   2415                   181             LIS   U1,5              LOOP COUNTER                 B3201440
 000160   F8F0 4F53 3332         182             LI    UF,C'OS32'        SET OS MASK                  B3201450
          0000 0166              183  XX.010    EQU   *                                              B3201460
 000166   D37E 0024              184             LB    U7STARTA,DIR.ATRB(UELINK)   GET ATTRIBUTE INFO B3201470
```

```
     INTERDATA 32-BIT BOOTSTRAP LOADER  **03-074M91R03A13**   PAGE   5   14:02:43  04/22/76

     BOOTSTRAP LOADER / RELOCATABLE SEGMENT 2

 00016A   C370 0010            185            THI    U7STARTA,DIRA.ACM  THIS ENTRY ACTIVE?                        B3201480
*00016E   233E                 186            BZ     XX.050             B IF NOT                                  B3201490
 000170   C370 00E0            187            THI    U7STARTA,X'E0'     CONTIG FILE?                              B3201500
*000174   213B                 188            BNZ    XX.050             B IF NOT                                  B3201510
 000176   55FE 0000            189            CL     UF,DIR.FNM(UELINK) IS THIS AN OS FILE?                       B3201520
*00017A   2138                 190            BNE    XX.050             IF NOT, BRANCH. (CHECK NEXT )             B3201530
 00017C   586E 0008            191            L      U6LOOP,DIR.EXT(UELINK)  GET FILE EXTENSION                   B3201540
 000180   1068                 192            SRLS   U6LOOP,8           POSITION RIGHT ONE BYTE                   B3201550
 000182   556D 0324            193            CL     U6LOOP,VERS(UDBIAS)   IS THIS DESIRED VERSION?              B3201560
 000186   433D 01B2            194            BE     XX.100(UDBIAS)     B IF OS FOUND                             B3201570
 00018A   CAE0 0030            195  XX.050    AHI    UELINK,48          BUMP FETCH POINTER                        B3201580
 00018E   2711                 196            SIS    U1,1               DEC ENTRY COUNTER                         B3201590
 000190   423D 0166            197            BNZ    XX.010(UDBIAS)     NOT DONE, CHECK MORE                      B3201600
 000194   58CD 0350            198            LDA    UCLSECN,VDBUF(UDBIAS)   DONE WITH THIS SECT. ANYMORE?        B3201610
 000198   423D 0152            199            BNZ    XX.000(UDBIAS)     YES, GET MORE                             B3201620
          0000 019C            200  XX.070    EQU    *                                                            B3201630
                               201  *  PROPER OS NOT IN DIRECTORY, LOOK IN VOL DESCRIPTOR                         B3201640
 00019C   4810 007E            202            LH     U1,X'7E'           EXTENSION ?                               B3201650
 0001A0   423D 02F0            203            BNZ    E.1(UDBIAS)        B IF YES                                  B3201660
 0001A4   58CD 0348            204            L      UCLSECN,OSP(UDBIAS)    GET POINTER TO OS                     B3201670
 0001A8   433D 02F0            205            BZ     E.1(UDBIAS)        B IF NO OS                                B3201680
 0001AC   588D 034C            206            L      U8ENDA,OSS(UDBIAS)                                           B3201690
*0001B0   2307                 207            B      XX.180             GO TO COMMON LOADER                       B3201710
                               208  *                                                                             B3201720
                               209  *  OS FOUND, DETERMINE SIZE                                                   B3201730
                               210  *                                                                             B3201740
          0000 01B2            211  XX.100    EQU    *                                                            B3201750
 0001B2   58CE 000C            212            LDA    UCLSECN,DIR.FLBA(UELINK)          PICK UP FILE 1ST LBA    B3201760
 0001B6   588E 0010            213            L      U8ENDA,DIR.LLBA(UELINK) PICK UP FILE EXTENT                  B3201780
 0001BA   0B8C                 214            SR     U8ENDA,UCLSECN                                               B3201783
 0001BC   26C1                 215            AIS    UCLSECN,1          BUMP PAST LIB BLOCK                       B3201786
          0000 01BE            216  XX.180    EQU    *                                                            B3201790
 0001BE   1188                 217            SLLS   U8ENDA,8           CONV LEN TO BYTES                         B3201800
 0001C0   0818                 218            LR     U1,U8ENDA                                                    B3201810
 0001C2   E6F0 1118            219            LDAI   UF,BOOTADR+SEGMENT3-SEGMENT2                                 B3201820
 0001C6   580F 0000            220  RELOCATE  LDA    U0,0(UF)           RELOCATE SEGMENT 2                        B3201830
 0001CA   5001 0000            221            STA    U0,0(U1)                                                     B3201840
 0001CE   5901 0000            222            C      U0,0(U1)           MOVING INTO GOOD MEMORY ?                 B3201850
 0001D2   423D 0308            223            BNE    E.7(UDBIAS)        B IF NO                                   B3201860
 0001D6   26F4                 224            AIS    UF,ADC                                                       B3201870
 0001D8   2614                 225            AIS    U1,ADC                                                       B3201880
 0001DA   C5F0 1278            226            CLHI   UF,LASTLOC+1-SEGMENT2+BOOTADR                                B3201890
 0001DE   208C                 227            BLS    RELOCATE                                                     B3201900
 0001E0   08D8                 228            LR     UDBIAS,U8ENDA                                                B3201910
*0001E2   CBD0 01E8            229            SAI    UDBIAS,SEGMENT3                                              B3201920
 0001E6   0308                 230            BR     U8ENDA             GO TO SEGMENT3                            B3201930
```

```
INTERDATA 32-BIT BOOTSTRAP LOADER  **03-074M91R03A13**   PAGE   6   14:02:44  04/22/76

BOOTSTRAP LOADER / RELOCATABLE SEGMENT 3

0001E8                          232           CNOP   4                                                    B3201950
       0000 01E8                233  SEGMENT3 EQU    *                                                    B3201960
0001E8 2781                     234           SIS    U8FNDA,1                                             B3201970
0001EA 0777                     235  READOS   XAR    U7STARTA,U7STARTA  SET OS IMAGE LOAD  ADR             B3201980
0001EC 41ED 01FC                236           BAL    UELINK,READ(UDBIAS) READ OS                          B3201990
0001F0 D1ED 0340                237           LM     UELINK,SAVE(UDBIAS)                                  B3202000
0001F4 D0E0 0078                238           STM    UELINK,X'78'                                         B3202010
0001F8 4300 0060                239           B      OSINITA            BRANCH TO THE OPERATING SYSTEM    B3202020
       0000 01FC                240  READ     EQU    *                                                    B3202030
0001FC DE2D 0328                241           OC     U2SELCH,SHSTOP(UDBIAS)                               B3202040
000200 080C                     242           LR     U0,UCLSECN                                           B3202044
000202 4D0D 4500 0330           243           DH     U0,SECCYL(UDBIAS,U5DEVCD)                            B3202046
000206 0891                     244           LR     U9CYLN,U1          0U0 = CYL                         B3202048
00020A 4D0D 4500 0338           245           DH     U0,SECTRK(UDBIAS,U5DEVCD)                            B3202050
000210 08A1                     246           LR     UAHEADN,U1         0U0 = TRK                         B3202052
000212 08B0                     247           LR     UBSECN,U0          REM = SECTOR                      B3202054
000214 0855                     248           LR     U5DEVCD,U5DEVCD                                      B3202056
000216 422D 024A                249           BP     RDSK2(UDBIAS)                                        B3202060


                                251  *----------------------------DIABLO DISK INPUT ROUTINE                B3202080


00021A DE3D 0328                253  RDSK1    OC     U3CTRL,CTRESET(UDBIAS) RESET CONTROLLER              B3202100
00021E 9D3F                     254  WAIT10   SSR    U3CTRL,UF          CTRL IDLE ?                       B3202110
*000220 2221                    255           BFC    2,WAIT10                                             B3202120
       0000 0222                256  WAIT15   EQU    *                                                    B3202130
000222 9D4F                     257           SSR    U4DEV,UF           DEVICE UNAVAILABLE?               B3202140
000224 421D 02F4                258           BTC    1,E.2(UDBIAS)      40MB DISC NOT AVAIL               B3202150
000228 C3F0 0010                259           THI    UF,X'10'                                             B3202160
*00022C 2035                    260           BNZ    WAIT15                                               B3202170
00022E 11A5                     261           SLLS   UAHEADN,5                                            B3202240
000230 06BA                     262           OR     UBSECN,UAHEADN                                       B3202245
000232 9849                     263  RDKLOOP1 WHR    U4DEV,U9CYLN       SEND CYLN NO TO DISK              B3202250
000234 DE4D 032B                264           OC     U4DEV,D1SEEK(UDBIAS) SEEK                            B3202260
000238 9D3F                     265  WAIT11   SSR    U3CTRL,UF          CONTROLLER IDLE ?                 B3202270
*00023A 2221                    266           BFC    2,WAIT11           NO,WAIT                           B3202280
00023C 9D4F                     267  WAIT12   SSR    U4DEV,UF           SEEK INCOMPLETE ?                 B3202290
00023E 427D 02F8                268           BTC    7,E.3(UDBIAS)      'SEEK INC' ERROR EXIT             B3202300
*000242 2083                    269           BTC    8,WAIT12           DISK NOT READY ?                  B3202310
000244 416D 0296                270           BAL    U6LOOP,DKCOMMON(UDBIAS)                              B3202320
000248 220B                     271           BFBS   0,RDKLOOP1                                           B3202330
```

```
INTERDATA 32-BIT BOOTSTRAP LOADER  **03-074M91R03A13**   PAGE   7   14:02:46  04/22/76

BOOTSTRAP LOADER / RELOCATABLE SEGMENT 3

                          273  *-----------------------------20M, 40M, 50M DISK INPUT ROUTINE          B3202350


         0000 024A        275  RDSK2     EQU    *                                                   B3202370
  00024A 9D4F             276               SSR    U4DEV,UF                                          B3202380
  00024C C3F0 0019        277               THI    UF,X'19'          SAFE ? READY ? ON-LINE ?        B3202390
  000250 423D 02FC        278               BNZ    E.4(UDBIAS)       40MB NT SAFE/NT RDY             B3202400
         0000 0254        279  RDKLOOP2  EQU    *                                                   B3202470
  000254 DE4D 032C        280               OC     U4DEV,D2RATTN(UDBIAS) RESET ATTENTION FLIP FLOP   B3202480
  000258 9D3F             281  WAIT21    SSR    U3CTRL,UF         CONTROLLER IDLE ?               B3202490
 *00025A 2221             282               BFC    2,WAIT21                                          B3202500
  00025C 9849             283               WHR    U4DEV,U9CYLN      SEND CYL NO                     B3202510
  00025E DE4D 032E        284               OC     U4DEV,D2SETCYL(UDBIAS) SET CYLINDER               B3202520
  000262 9D3F             285  WAIT22    SSR    U3CTRL,UF         CTRL IDLE ?                     B3202530
 *000264 2221             286               BFC    2,WAIT22                                          B3202540
  000266 DE4D 032D        287               OC     U4DEV,D2RHEAD(UDBIAS)  CLEAR HEAD REG.            B3202550
  00026A 9D3F             288  WAIT23    SSR    U3CTRL,UF         CTRL IDLE ?                     B3202560
 *00026C 2221             289               BFC    2,WAIT23          NO,WAIT                         B3202570
  00026E 984A             290               WHR    U4DEV,UAHEADN     SEND HEAD NO                    B3202580
  000270 DE4D 032F        291               OC     U4DEV,D2SETHED(UDBIAS)  ST HEAD                   B3202590
  000274 9D3F             292  WAIT24    SSR    U3CTRL,UF         CTRL IDLE ?                     B3202600
 *000276 2221             293               BFC    2,WAIT24                                          B3202610
  000278 DE4D 032B        294               OC     U4DEV,D2SEEK(UDBIAS) SEEK                         B3202620
  00027C 9D3F             295  WAIT25    SSR    U3CTRL,UF         CTRL IDLE ?                     B3202630
 *00027E 2221             296               BFC    2,WAIT25                                          B3202640
  000280 9D4F             297  WAIT26    SSR    U4DEV,UF                                          B3202650
 *000282 2081             298               BTC    8,WAIT26                                          B3202660
  000284 C3F0 0053        299               THI    UF,X'53'          ERROR ?                         B3202670
  000288 423D 0300        300               BNZ    E.5(UDBIAS)       40MB UNSAFE/SK INC/NT RDY       B3202680
  00028C 08FA             301               LDAR   UF,UAHEADN                                        B3202690
  00028E 11FA             302               SLLS   UF,10                                             B3202700
  000290 06F9             303               OAR    UF,U9CYLN                                         B3202710
  000292 E66D 0254        304               LDAI   U6LOOP,RDKLOOP2(UDBIAS)                           B3202720
```

BOOTSTRAP LOADER / RELOCATABLE SEGMENT 3

```
                          306  *----------------------------DISK COMMON ROUTINE                                  B3202740


         0000 0296        308  DKCOMMON EQU   *                                                                 B3202760
 000296  3477             309           EXHR  U7STARTA,U7STARTA                                                 B3202770
 000298  9A27             310           WDR   U2SELCH,U7STARTA                                                  B3202780
 00029A  3477             311           EXHR  U7STARTA,U7STARTA                                                 B3202790
 00029C  9827             312           WHR   U2SELCH,U7STARTA                                                  B3202800
 00029E  3488             313           EXHR  U8ENDA,U8ENDA       SEND ENDA TO SELCH                            B3202810
 0002A0  9A28             314           WDR   U2SELCH,U8ENDA                                                    B3202820
 0002A2  3488             315           EXHR  U8ENDA,U8ENDA                                                     B3202830
 0002A4  9828             316           WHR   U2SELCH,U8ENDA                                                    B3202840
 0002A6  0855             317           LR    U5DEVCD,U5DEVCD     40 MB OR BIGGER                             B3202850
*0002A8  2124             318           BP    COMN1               B IF YES                                    B3202860
 0002AA  9849             319           WHR   U4DEV,U9CYLN                                                    B3202870
 0002AC  9A3B             320           WDR   U3CTRL,UBSECN                                                   B3202880
*0002AE  2303             321           B     COMN2                                                           B3202890
 0002B0  9A3B             322  COMN1    WDR   U3CTRL,UBSECN                                                   B3202900
 0002B2  983F             323           WHR   U3CTRL,UF                                                       B3202910
         0000 02B4        324  COMN2    EQU   *                                                               B3202920
 0002B4  DE3D 032A        325           OC    U3CTRL,CTREAD(UDBIAS)     READ -> CTRL                        B3202930
 0002B8  DE2D 0329        326           OC    U2SELCH,SHGORD(UDBIAS)    GO & READ -> SELCH                  B3202940
 0002BC  9D2F             327  WAIT31   SSR   U2SELCH,UF              SELCH BUSY ?                          B3202950
*0002BE  2081             328           BTC   8,WAIT31                                                        B3202960
 0002C0  DE2D 0328        329           OC    U2SELCH,SHSTOP(UDBIAS)    STOP SELCH                          B3202970
 0002C4  9B20             330           RDR   U2SELCH,U0            READ FIRST BYTE OF END ADR (0:23<-0)      B3202980
 0002C6  9921             331           RHR   U2SELCH,U1            READ THE RESTR OF END ADR (0:16 <-0)      B3202990
 0002C8  3400             332           EXHR  U0,U0                 SHIFT U0 LEFT BY 1 HW                     B3203000
 0002CA  0601             333           OAR   U0,U1                 GET END ADR                               B3203010
 0002CC  DE2D 0328        334           OC    U2SELCH,SHSTOP(UDBIAS)                                          B3203020
 0002D0  9D3F             335  WAIT32   SSR   U3CTRL,UF              CTRL IDLE ?                           B3203030
*0002D2  2221             336           BFC   2,WAIT32                                                        B3203040
 0002D4  421D 0304        337           BTC   1,E.6(UDBIAS)         CONT DAT XFR ERR                          B3203050
 0002D8  C3F0 0010        338           THI   UF,X'10'              CYLINDER OVERFLOW                         B3203060
 0002DC  033E             339           BZR   UELINK                NO, RETURN                                B3203070
 0002DE  0B07             340           SAR   U0,U7STARTA           U0 = LENGTH OF DATA BLK READ - ADC        B3203080
 0002E0  2604             341           AIS   U0,ADC                                                          B3203090
 0002E2  C400 FF00        342           NHI   U0,X'FF00'            ADJUST TO 256-BYTE BOUNDARY               B3203100
 0002E6  0A70             343           AAR   U7STARTA,U0           GET NEW START ADR                         B3203110
 0002E8  2691             344           AIS   U9CYLN,1              ADD CYL NO BY 1                           B3203120
 0002EA  07AA             345           XAR   UAHEADN,UAHEADN                                                 B3203130
 0002EC  07BB             346           XAR   UBSECN,UBSECN                                                   B3203140
 0002EE  0306             347           BR    U6LOOP                                                          B3203150
```

```
INTERDATA 32-BIT BOOTSTRAP LOADER  **03-074M91R03A13**   PAGE   9   14:02:48  04/22/76

BOOTSTRAP LOADER / ERROR EXITS

 0002F0   2411                 349  E.1       LIS    U1,1               OS NOT FOUND                  B3203170
 0002F2   230C                 350             BS     ER.XIT                                           B3203180
 0002F4   2412                 351  E.2       LIS    U1,2               DISC NOT AVAIL                B3203190
 0002F6   230A                 352             BS     ER.XIT                                           B3203200
 0002F8   2413                 353  E.3       LIS    U1,3               SK NT CMP/DSC NT RDY          B3203210
 0002FA   2308                 354             BS     ER.XIT                                           B3203220
 0002FC   2414                 355  E.4       LIS    U1,4               40MB NT SAFE/NT ON LINE       B3203230
 0002FE   2306                 356             BS     ER.XIT                                           B3203240
 000300   2415                 357  E.5       LIS    U1,5               40MB UNSF/SK INC              B3203250
 000302   2304                 358             BS     ER.XIT                                           B3203260
 000304   2416                 359  E.6       LIS    U1,6               DAT XFR ERR                   B3203270
 000306   2302                 360             BS     ER.XIT                                           B3203280
 000308   2417                 361  E.7       LIS    U1,7               OS TO BIG FOR MEMORY          B3203290
 00030A   2401                 362  ER.XIT    LIS    U0,1                                             B3203300
 00030C   DE0D 0328            363             OC     U0,SHSTOP(UDBIAS)  DISPLAY TO NORMAL MODE        B3203310
 000310   9A01                 364             WDR    U0,U1                                            B3203320
 000312   DE0D 0328            365             OC     U0,SHSTOP(UDBIAS)  DISPLAY TO NORMAL MODE        B3203330
 000316   D1ED 0340            366  ERR.END   LM     UELINK,SAVE(UDBIAS)                              B3203340
 00031A   D0E0 0078            367             STM    UELINK,X'78'                                     B3203350
 00031E   110F                 368             SLLS   U0,15                                            B3203360
          0000 0320            369  HANGWAIT  EQU    *                                                B3203370
 000320   9510                 370             EPSR   U1,U0              ENTER WAIT STATE              B3203380
*000322   2201                 371             B      HANGWAIT                                         B3203390
```

```
    INTERDATA 32-BIT BOOTSTRAP LOADER  **03-074M91R03A13**   PAGE  10   14:02:48  04/22/76

    BOOTSTRAP LOADER / ERROR EXITS

000324                          373             CNOP    4                                       B3203410
000324  0000 0000              374  VERS       DC      0                                       B3203420
000328  4630                    375  SHSTOP     DC      X'4830'         SELCH STOP COMMAND      B3203430
        0000 0329               376  SHGORD     EQU     SHSTOP+1        SELCH READ/GO COMMAND   B3203440
        0000 0328               377  CTRESET    EQU     SHSTOP          CONTROLLER RESET COMMAND B3203450
00032A  C1C2                    378  CTREAD     DC      X'C1C2'                                 B3203460
                                379  * DISC COMMANDS FOR 2.5-10 AND 40 MB FILES RESPECTIVELY     B3203470
        0000 032B               380  D1SEEK     EQU     CTREAD+1                                B3203480
        0000 032B               381  D2SEEK     EQU     D1SEEK                                  B3203490
00032C  C8C4                    382  D2RATTN    DC      X'C8C4'                                 B3203500
        0000 032D               383  D2RHEAD    EQU     D2RATTN+1                               B3203510
00032E  D0E0                    384  D2SETCYL   DC      X'D0E0'                                 B3203520
        0000 032F               385  D2SETHED   EQU     D2SETCYL+1                              B3203530
000330  0030                    386  SECCYL     DC      H'48'           <= 10                   B3203540
000332  0190                    387             DC      H'400'          40                      B3203542
000334  0140                    388             DC      H'320'          67                      B3203544
000336  04C0                    389             DC      H'1216'         252                     B3203546
000338  0018                    390  SECTRK     DC      H'24'           <= 10                   B3203548
00033A  0014                    391             DC      H'20'           40                      B3203550
00033C  0040                    392             DC      H'64'           67                      B3203552
00033E  0040                    393             DC      H'64'           252                     B3203554
        0000 0270               394  LENGTH     EQU     *-SEGMENT2                              B3203570
                                395  * VOLUME DESCRIPTOR PARAMETERS INCLUDING OS SIZE AND LOGICAL SECTOR B3203580
000340                          396             CNOP    4                                       B3203590
        0000 0340               397  SAVE       EQU     *                                       B3203600
        0000 0347               398  LASTLOC    EQU     SAVE+7          LAST LOCATION TO RELOCATE B3203610
        0000 0348               399  OSP        EQU     SAVE+8                                  B3203620
        0000 034C               400  OSS        EQU     OSP+4                                   B3203630
        0000 0350               401  VDBUF      EQU     OSS+4                                   B3203640
        0000 044F               402  VDBUFE     EQU     VDBUF+255                               B3203650
        0000 0350               403  VD.VOL     EQU     VDBUF                                   B3203660
        0000 0354               404  VD.ATRB    EQU     VD.VOL+4                                B3203670
        0000 0358               405  VD.FDP     EQU     VD.ATRB+4                               B3203680
        0000 035C               406  VD.OSP     EQU     VD.FDP+4                                B3203690
        0000 0360               407  VD.OSS     EQU     VD.OSP+4                                B3203700
        0000 0364               408  VD.MAP     EQU     VD.OSS+4                                B3203710
000340                          409             END                                             B3203720
```

```
    INTERDATA 32-BIT BOOTSTRAP LOADER  **03-074M91R03A13**   PAGE  11   14:02:49  04/22/76

    BOOTSTRAP LOADER / ERROR EXITS

NO ERRORS     3 SQUEZ PASSES

CAL 04-00

  ABSTOP     0000 0340
  ADC        0000 0004              224   225   341
  BOOTADR    0000 1000              124   127   128   142   219   226
  BOOTLDR    0000 0080
  BOOTSTRP   0000 00A6
  BOOTSTRT   0000 00DE              124
  BOSTRT1    0000 0134              163
  BOTRANGE   0000 00C8              123
  CNV        0000 00D0              124   147   151   155
  COMN1      0000 02B0              318
  COMN2      0000 02B4              321
  CTREAD     0000 032A              325   380
  CTRESET    0000 0328              253
  CTRLA      0000 007C              158
  D1SEEK     0000 032B              264   381
  D2RATTN    0000 032C              280   383
  D2RHEAD    0000 032D              287
  D2SEEK     0000 032B              294
  D2SETCYL   0000 032E              284   385
  D2SETHED   0000 032F              291
  DEVA       0000 007A              159
  DEVCD      0000 007B              161
  DIR        0000 0030
  DIR.ATRB   0000 0024              184
  DIR.BKSZ   0000 0025
  DIR.CSEC   0000 0028
  DIR.DATE   0000 0018
  DIR.EXT    0000 0008               58   191
  DIR.FLBA   0000 000C              212
  DIR.FNM    0000 0000              189
  DIR.INBS   0000 0026
  DIR.KEYS   0000 0014
  DIR.LLBA   0000 0010              213
  DIR.LRCL   0000 0016
  DIR.LUSE   0000 001C
  DIR.RCNT   0000 0022
  DIR.RKEY   0000 0015
  DIR.RSAS   0000 00F4
  DIR.VERS   0000 000B
  DIR.WCNT   0000 0020
  DIR.WKEY   0000 0014
  DIRA.ACB   0000 0003
  DIRA.ACM   0000 0010              185
  DIRA.ALB   0000 0007
  DIRA.ALM   0000 0001
  DKCOMMON   0000 0296              270
  E.1        0000 02F0              203   205
  E.2        0000 02F4              258
  E.3        0000 02F8              268
  E.4        0000 02FC              278
```

BOOTSTRAP LOADER / ERROR EXITS

E.5	0000	0300	300										
E.6	0000	0304	337										
E.7	0000	0308	223										
ER.XIT	0000	030A	350	352	354	356	358	360					
ERR.END	0000	0316											
HANGWAIT	0000	0320	371										
IMPTOP	0000	0000I											
INIT	0000	00A2	119										
INITADR	0000	009E	112	113									
LADC	0000	0002											
LASTLOC	0000	0347	226										
LENGTH	0000	0270	127										
OSINITA	0000	0060	239										
OSP	0000	0348	172	204	400								
OSS	0000	034C	206	401									
PURETOP	0000	0000P											
R1	0000	0001	111	114	115								
R2	0000	0002											
R3	0000	0003	112	116									
R4	0000	0004	113	117									
RDKLOOP1	0000	0232	271										
RDKLOOP2	0000	0254	304										
RDSK1	0000	021A											
RDSK2	0000	024A	249										
READ	0000	01FC	170	179	236								
READOS	0000	01EA											
RELOCATE	0000	01C6	227										
S1END	0000	1270	129										
SAVE	0000	0340	144	237	366	398	399						
SECCYL	0000	0330	243										
SECTRK	0000	0338	245										
SEGMENT2	0000	00D0	142	219	226	394							
SEGMENT3	0000	01E8	219	229									
SELCHA	0000	007D	157										
SHGORD	0000	0329	326										
SHSTOP	0000	0328	241	329	334	363	365	376	377				
U0	0000	0000	220	221	222	242	243	245	247	330	332	332	333	340	341
			342	343	362	363	364	365	368	370					
U1	0000	0001	181	196	202	218	221	222	225	244	246	331	333	349	351
			353	355	357	359	361	364	370						
U2SELCH	0000	0002	157	241	310	312	314	316	326	327	329	330	331	334	
U3CTRL	0000	0003	158	253	254	265	281	285	288	292	295	320	322	323	325
			335												
U4DEV	0000	0004	122	123	159	257	263	264	267	276	280	283	284	287	290
			291	294	297	319									
U5DEVCD	0000	0005	160	165	243	245	248	248	317	317					
U6LOOP	0000	0006	191	192	193	270	304	347							
U7STARTA	0000	0007	167	178	184	185	187	235	235	309	309	310	311	311	312
			340	343											
U8ENDA	0000	0008	168	206	213	214	217	218	228	230	234	313	313	314	315
			315	316											
U9CYLN	0000	0009	244	263	283	303	319	344							
UAHEADN	0000	000A	246	261	262	290	301	345	345						
UBSECN	0000	000B	247	262	320	322	346	346							

INTERDATA 32-BIT BOOTSTRAP LOADER **03-074M91R03A13** PAGE 13 14:02:58 04/22/76

BOOTSTRAP LOADER / ERROR EXITS

UCLSECN	0000 000C	169 169 176 198 204 212 214 215 242
UDBIAS	0000 000D	142 144 147 148 151 152 155 156 167 168 170 171 172
		176 177 178 179 180 193 194 197 198 199 203 204 205
		206 223 228 229 236 237 241 243 245 249 253 258 264
		268 270 278 280 284 287 291 294 300 304 325 326 329
		334 337 363 365 366
UELINK	0000 000E	138 140 143 144 147 151 155 170 171 172 179 180 184
		189 191 195 212 213 236 237 238 339 366 367
UF	0000 000F	136 137 139 145 146 148 149 150 152 153 154 156 161
		162 164 165 182 189 219 220 224 226 254 257 259 265
		267 276 277 281 285 288 292 295 297 299 301 302 303
		323 327 335 338
VD.ATRB	0000 0354	405
VD.FDP	0000 0358	176 406
VD.MAP	0000 0364	
VD.OSP	0000 035C	171 407
VD.OSS	0000 0360	408
VD.VOL	0000 0350	404
VDBUF	0000 0350	167 178 180 198 402 403
VDBUFE	0000 044F	168
VERS	0000 0324	148 152 156 193
WAIT10	0000 021E	255
WAIT11	0000 0238	266
WAIT12	0000 023C	269
WAIT15	0000 0222	260
WAIT21	0000 0258	282
WAIT22	0000 0262	286
WAIT23	0000 026A	289
WAIT24	0000 0274	293
WAIT25	0000 027C	296
WAIT26	0000 0280	298
WAIT31	0000 02BC	328
WAIT32	0000 02D0	336
XX.000	0000 0152	199
XX.010	0000 0166	197
XX.050	0000 018A	186 188 190
XX.070	0000 019C	177
XX.100	0000 01B2	194
XX.180	0000 01BE	207

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

_____

_____

FOLD FOLD

Check the appropriate item.

☐ Error Page No. _____ Drawing No. _____

☐ Addition Page No. _____ Drawing No. _____

☐ Other Page No. _____ Drawing No. _____

Explanation:

FOLD FOLD

CUT ALONG LINE

Fold and Staple

No postage necessary if mailed in U.S.A.

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT No. 22
OCEANPORT, N. J.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY:

INTERDATA®

Subsidiary of PERKIN-ELMER
Oceanport, New Jersey 07757, U.S.A.

TECH PUBLICATIONS DEPT. MS 229

FOLD FOLD

STAPLE STAPLE